AD-A095 188 BELL HELICOPTER TEXTRON FORT WORTH TX F/G 9/2
THE DATA FROM AEROMECHANICS TEST AND ANALYTICS -- MANAGEMENT AN--ETC(U)
DEC 80 R B PHILBRICK DAAK51-79-C-0015
UNCLASSIFIED BHT-699-099-025-VOL-1 USAAVRADCOM-TR-80-D-30A NL

1 OF 3
AD
A095

USAAVRADCOM-TR-80-D-30A

LEVEL

AD A095188

THE DATA FROM AEROMECHANICS TEST AND ANALYTICS — MANAGEMENT AND ANALYSIS PACKAGE (DATAMAP)
Volume I - User's Manual

Richard B. Philbrick
BELL HELICOPTER TEXTRON
P. O. Box 482
Fort Worth, Tex. 76101

December 1980

FEB 19 1981

A

Final Report

Approved for public release;
distribution unlimited.

Prepared for

APPLIED TECHNOLOGY LABORATORY
U. S. ARMY RESEARCH AND TECHNOLOGY LABORATORIES (AVRADCOM)
Fort Eustis, Va. 23604

81 2 17 063

APPLIED TECHNOLOGY LABORATORY POSITION STATEMENT

This report has been reviewed by the Applied Technology Laboratory, US Army Research and Technology Laboratories (AVRADCOM), and is considered to be technically sound.

DATAMAP is a computer software system which provides direct access to large data bases, performs analysis and derivations, and provides the user with various options for output display, interactively or through batch processing. DATAMAP was designed to utilize the AH-1G Operational Loads Survey data but is general enough to accommodate other large, time-based, data sets resulting from test or analysis.

Improvements have been made to DATAMAP to enhance its graphics, analysis, and operational capabilities in order to expand its versatility and usefulness as an engineering tool. Volume I of this report explains the general structure and capabilities of the improved DATAMAP, and Volume II provides information on programming considerations.

This project was conducted under the technical management of D. J. Merkley of the Aeronautical Technology Division.

19   12   284

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 USAAVRADCOM TR-80-D-30A	2. GOVT ACCESSION NO. AD-A095188	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 THE DATA FROM AEROMECHANICS TEST AND ANALYTICS - MANAGEMENT AND ANALYSIS PACKAGE (DATAMAP). Volume I. User's Manual.		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report.
	14	6. PERFORMING ORG. REPORT NUMBER BHT-699-099-025-VOL-1
7. AUTHOR(s) 10 Richard B. Philbrick	15	8. CONTRACT OR GRANT NUMBER(s) DAAK51-79-C-0015
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bell Helicopter Textron P. O. Box 482 Fort Worth, Texas 76101	16	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 612209 1L162209AH76 00 265 EK
11. CONTROLLING OFFICE NAME AND ADDRESS Applied Technology Laboratory, US Army Research & Technology Laboratories (AVRADCOM) Fort Eustis, Virginia 23604	11	12. REPORT DATE December 1980
		13. NUMBER OF PAGES 278
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 17 00		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Volume I of a two-volume report.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Helicopters, Data Bases, Data Reduction, Data Management, Computer Graphics, Interactive Graphics, Signal Processing, Mathematical Analysis, Acoustic Analysis.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Data from Aeromechanics Test and Analytics - Management and Analysis Package (DATAMAP) was designed and programmed as a computer software tool for data management and processing of large, time-based data bases. Particular attention is given to rotorcraft-related analyses. The package will process data stored in two basic formats. The first format is that used for the Operational Loads Survey (OLS) test data base and anticipated for use in planned flight test programs. The second format is more → cont.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

054200 JW

20. ABSTRACT (Continued)

general; it accommodates various data structures common to analytical data bases. This particular input capability is demonstrated by an interface between the Rotorcraft Flight Simulation Program (C81) and DATAMAP. The package transfers selected data to a large, direct access disc file and maintains the data on a semi-permanent basis. Data are retrieved from this file, processed, and displayed interactively or in batch. Plot output is generated on a Tektronix 4014 or an incremental plotter (e.g., Calcomp).

A small sample of available processing options includes amplitude spectrum, harmonic analysis, digital filtering, auto-spectral density, frequency response function, acoustic analyses, and blade static pressure and normal force coefficient derivations. This program will accommodate data from multiple sensors simultaneously for processing of functions with two geometric independent variables (e.g., chord and radius). Output options include printout, single or multiple curve X-Y plots, contour plots, and pictorial representation of 3-dimensional surfaces. User input is free field with errors diagnosed where possible. Prompting for available command input is optional.

The package is written entirely in FORTRAN. Package specifications require nonstandard FORTRAN coding to be used, but the package has been made as easily transportable as possible. In particular, DATAMAP is installed on a Digital Equipment Corporation VAX 11/780 as well as on the originally intended IBM 360 and 370 systems.

This report is in two volumes. Volume I is a user's manual and Volume II is a systems manual for assistance in program maintenance, modification, and/or installation.

PREFACE

DATAMAP Version 2.00 (formerly known as the Operational Loads Survey Data Management System) was developed under Contract DAAJ02-77-C-0053 for the Applied Technology Laboratory, U. S. Army Research and Technology Laboratories (AVRADCOM) to process the data measured by the AH-1G Helicopter Aerodynamic and Structural Loads Survey (OLS). This survey, performed for Applied Technology Laboratory (ATL), resulted in a comprehensive base of helicopter test data including measurements such as airfoil surface pressures, leading-edge stagnation point, local flow magnitude and direction, blade accelerations, bending moments, and the attendant responses in the control system and airframe. The data base included 72,000 separate digitized functions of time from 367 transducers. This data base, together with the techniques used for measurement of the data, is documented in Reference 1.

DATAMAP Version 2.00, documented in Reference 2, has been successfully used to process the OLS data base for a number of projects both at ATL (Fort Eustis, Virginia) and at Bell Helicopter Textron (BHT). To enhance the usefulness of this software, Contract DAAK51-79-C-0015 was awarded in May of 1979 by ATL. This contract required BHT to modify DATAMAP to incorporate new analysis, operational, and graphic capabilities, and also to provide an interface between the Rotorcraft Flight Simulation Program (C81) and DATAMAP. Documentation prepared under this contract for DATAMAP consists of two volumes. Volume I provides user instructions and information

---

[1]Gerald A. Shockey, Joe W. Williamson, and Charles R. Cox, AH-1G HELICOPTER AERODYNAMIC AND STRUCTURAL LOADS SURVEY, Bell Helicopter Textron, USAAMRDL Technical Report 76-39, Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Va., February 1977, AD A036910.

[2]Richard B. Philbrick, and Alfred L. Eubanks, OPERATIONAL LOADS SURVEY - DATA MANAGEMENT SYSTEM, Bell Helicopter Textron, USARTL Technical Report 78-52A and 52B, Applied Technology Laboratory, Fort Eustis, Virginia, January 1979, AD A065129 and AD A065270.

[3]James R. Van Gaasbeek, and P. Y. Hsieh, ROTORCRAFT FLIGHT SIMULATION PROGRAM C81 WITH DATAMAP INTERFACE, Volumes I and II, Bell Helicopter Textron, USAAVRADCOM Technical Report 80-D-38A and 80-D-38B, Applied Technology Laboratory, U.S. Army Research and Technology Laboratories, Fort Eustis, Virginia,

on the analytical methods used in the software. Volume II, the Systems Manual, details the various programming considerations. Information on the C81 - DATAMAP interface is also contained in Reference 3, which was produced for this contract.

Technical program direction was provided by Mr. D. J. Merkley of ATL. Principal Bell Helicopter Textron personnel associated with the current contract were Messrs. R. B. Philbrick, A. L. Eubanks, W. R. Dodds, J. R. Van Gaasbeek, and P. Y. Hsieh.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Concluded)

LIST OF TABLES

1. INTRODUCTION

1.1 DATAMAP HISTORY AND NEW SYSTEM REQUIREMENTS

The Data from Aeromechanics Test and Analytics - Management and Analysis Package (DATAMAP Version 2.00, as documented in Reference (2)) has proved to be a useful tool for processing the Operational Loads Survey (OLS) test data base and certain other test data. DATAMAP provides access, processing, and graphic or printed presentation of helicopter test data while operating in a batch or interactive mode. The package was written for installation on IBM 360 and 370 series computers. However, the package was also written to be reasonably transportable and has been successfully installed on a Digital Equipment Corporation VAX 11/780 computer.

The requirements of Contract DAAK51-79-C-0015 provide for specific improvements in the analytical, graphic, and operational capabilities of the package. In addition, a capability is required to process input in a general format to which test or analytic data bases can be readily converted. As a specific realization of this capability, an interface to allow Rotorcraft Flight Simulation Program (C81) data to be read and processed by DATAMAP is required.

1.2 GENERAL SYSTEM CAPABILITIES

DATAMAP Version 3.00 is the product of this contractual effort (all subsequent references to DATAMAP will assume Version 3.00). A general format that accommodates many of the special characteristics of both test and analytic time based data has been specified, and DATAMAP has been converted to read this format. This capability allows test and analytic data to be displayed in a uniform graphic format and, in many cases, as curves on the same plot. The capability to read BHT-Ground Data Center (BHT-GDC) format data tapes (e.g., OLS data tapes) is retained. In addition, the analytic, graphic, and operational capabilities are greatly expanded so that the package is easier to use, more powerful, and more generally applicable.

DATAMAP capabilities are described briefly in this section under the categories of data storage/retrieval, analysis, data presentation, and ease of use. Then, an example of the capabilities of DATAMAP is described in general terms. DATAMAP reads either BHT-GDC format digital data tapes, or other time-based data stored in a specified general format that is called the Data Transfer File (DTF) format. DATAMAP will transfer user-selected data from the specified source, to a direct access disc file called the Master File. A directory in the Master File allows automatic, rapid retrieval of the data by

the processing/display part of the system, which can run in interactive or batch modes. Measured or analytic data storage and retrieval features include the following:

- Data are stored on a direct access file with a directory for rapid retrieval.

- The sampling rate for BHT-GDC format data can be reduced in the storage process.

- The sampling rate for DTF format data can be changed arbitrarily.

- Data can be filtered in the storage process.

- There is no program limit to the number of data streams or the length of data streams that can be stored.

- Multiple users can access the data simultaneously.

The DTF format is designed to accommodate many of the different structural features of time-based data bases so that conversion of data to this format should be a reasonably simple task. In particular, this format will accommodate variable or constant sample rates and data organized in series or parallel. A parallel data organization implies that all channel output values for a particular time instant appear together. A series organization implies that all of the output for one channel for a record is stored contiguously. The DTF format is also compatible with a mixture of parallel and series organization.

Various analyses may be performed on the basic measured or analytic parameters contained in the data base and, in addition, several parameters may be derived from basic parameters. The computational capabilities available to the user in the DATAMAP Processing Program are detailed in Tables 1 and 2. The list of available analysis options is considerably expanded for DATAMAP Version 3.00. These analyses and derivations can be performed in multiple dimensions (e.g., time, chord, and radius). Sequences of analyses and/or derivations can be performed on a set of data. For example, Main Rotor Shaft Horsepower could be derived from measured parameters and then the result filtered.

Measured parameters and process outputs can be presented in various ways as shown in Table 3. Simple X-Y plots or multiple-curve X-Y plots are available. Three-dimensional outputs in the form of contour plots and surface perspective drawings in rectangular and cylindrical coordinate systems are available. These output options are available on a Tektronix 4014

Table 1. ANALYSIS OPTIONS

AMPLITUDE SPECTRA	STOCHASTIC PROCESS ANALYSES:
HARMONIC ANALYSIS	• FREQUENCY RESPONSE FUNCTION
DIGITAL FILTERING	• COHERENCE FUNCTION
MOVING BLOCK DAMPING	• AUTO-SPECTRAL DENSITY
CYCLE AVERAGING	• CROSS-SPECTRAL DENSITY
MIN/MAX ANALYSIS	• AUTO-CORRELATION
ACOUSTIC ANALYSES:	• CROSS-CORRELATION
• NARROW BAND ANALYSIS	BASIC STATISTICAL ANALYSES:
• OCTAVE ANALYSIS	• MEAN
• THIRD OCTAVE ANALYSIS	• VARIANCE
• PERCEIVED NOISE LEVEL CALCULATION	• STANDARD DEVIATION
• A, B, C, AND D NETWORK WEIGHTED INTEGRATION	• CHI-SQUARE TEST FOR NORMAL DISTRIBUTION

Table 2. DERIVED PARAMETER OPTIONS

TRUE AIRSPEED	BLADE STATIC PRESSURE COEFFICIENT
ROTOR AZIMUTH	BLADE NORMAL FORCE COEFFICIENT
ROTOR RPM	BLADE CHORDWISE FORCE COEFFICIENT
SHAFT HORSEPOWER	BLADE PITCHING MOMENT COEFFICIENT
SHAFT THRUST COEFFICIENT	BLADE DISPLACEMENT
SHAFT TORQUE COEFFICIENT	BLADE LOCAL FLOW MAGNITUDE
DENSITY ALTITUDE	BLADE LOCAL FLOW DIRECTION

Table 3. OUTPUT CAPABILITIES AND FEATURES

PRINTOUT:	ANY OUTPUT MAY BE PRINTED
X-Y PLOTS:	MULTIPLE CURVES ON ONE PLOT
	LINEAR OR LOG SCALES IN EITHER DIRECTION
	SMALL DATA 'WINDOWS' CAN BE SELECTED
	TEKTRONIX GRAPHICS CURSOR CAN BE ACCESSED TO EVALUATE POINTS
CONTOUR PLOTS:	RECTANGULAR OR CYLINDRICAL FORMAT
	CONTOUR INTERVAL, INITIAL CONTOUR LEVEL, AND CONTOUR DENSITY ARE SELECTABLE
SURFACE PLOTS:	RECTANGULAR OR CYLINDRICAL FORMAT
	SURFACE MAY BE VIEWED FROM ANY ANGLE
GENERAL:	ANY GRAPHIC OUTPUT CAN BE TO TEKTRONIX OR CALCOMP EMULATING DEVICE
	TEKTRONIX PLOTS MAY BE EXPANDED TO VIEW ONLY THE PLOTTING AREA
	CONVERSION OF DEPENDENT VARIABLE UNITS IS AVAILABLE

terminal in the Interactive Graphics mode of operation or on an off-line plotter (e.g., CALCOMP or Houston Instrument DP-1) in the batch mode. Printed listings of output are available in either mode.

User command input for both the transfer of data to the Master File and the processing of data from the Master File is free field. For transfer of data to the Master File, the run is assumed to be batch mode; however, an interactive utility program exists for creating the input for this process. For the retrieval and processing of data from the Master File, the input can be interactive or batch. Features that assist the user in generation of command input for processing include:

- The input format is free field.
- Many types of errors are detected in the input process.
- There are defaults for some entries.
- The program can list the available options, defaults, and/or meaning for each input entry.
- The assistance described in the item above is optional.
- Sequences of commands can be built, stored and executed by name.

Correlation of the data with a rotor azimuth pulse train is available both as part of processing and output scale generation. Most of the derivations and some of the analysis procedures require this azimuth pulse train to be present. However, other processing capabilities, data management features, and the graphics functions can readily be used for nonhelicopter applications.

As an example of application of this system, the user loads the Master File with absolute pressure measurements along with the measured parameters: Main Rotor Azimuth, True Indicated Airspeed, Boom System Static Pressure, and Outside Air Temperature. In addition, an analytic program is used to supply normal force coefficient data from a simulation of the flight condition flown to generate the measured data. Rotorcraft Flight Simulation Program (C81) simulation data are used in this example. The simulation output data are loaded on the Master File as well, using the DTF input option. Then the Processing Program is used in the interactive graphics mode to produce Figures 1 through 12.

Figure 1 shows the blade absolute pressure for all the sensors on the top surface at the 85-percent radial station as a function of blade azimuth position in degrees. The data shown

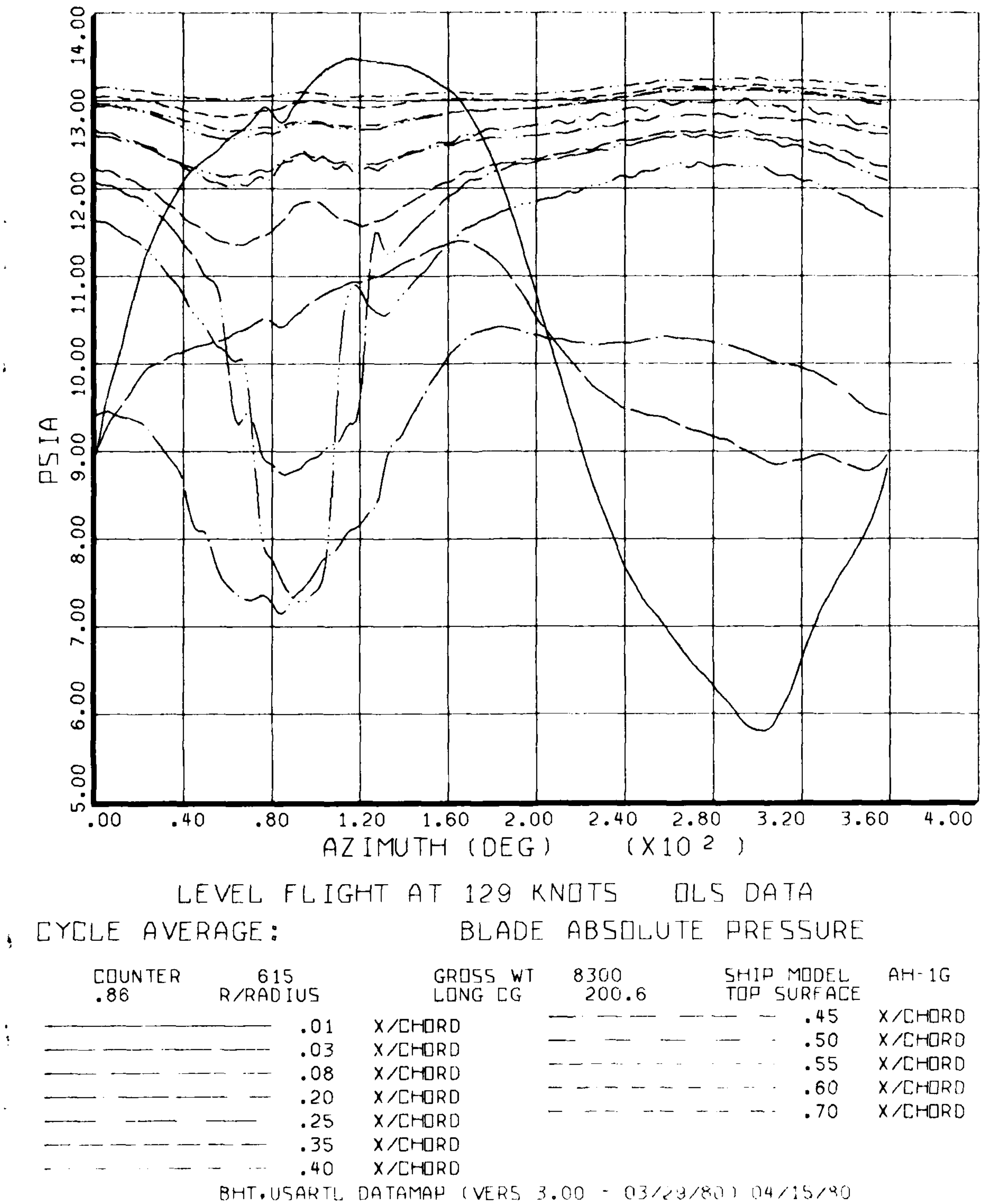


Figure 1. Cycle-averaged blade absolute pressure for one radial position.

have been cycle-averaged. That is, a representative history of data spanning one rotor cycle has been synthesized by averaging together data spanning several contiguous cycles. Figure 2 shows a contour plot of blade absolute pressure on the top surface of the blade for a fixed azimuth position. Figure 3 shows the same data as a pictorial depiction of a function of two independent variables or "surface plot." Figure 4 shows a surface plot of all the top surface sensors at the 1-percent chord position. Figure 5 is a contour plot of the same data. Both of these plots are in cylindrical format to better represent blade azimuth.

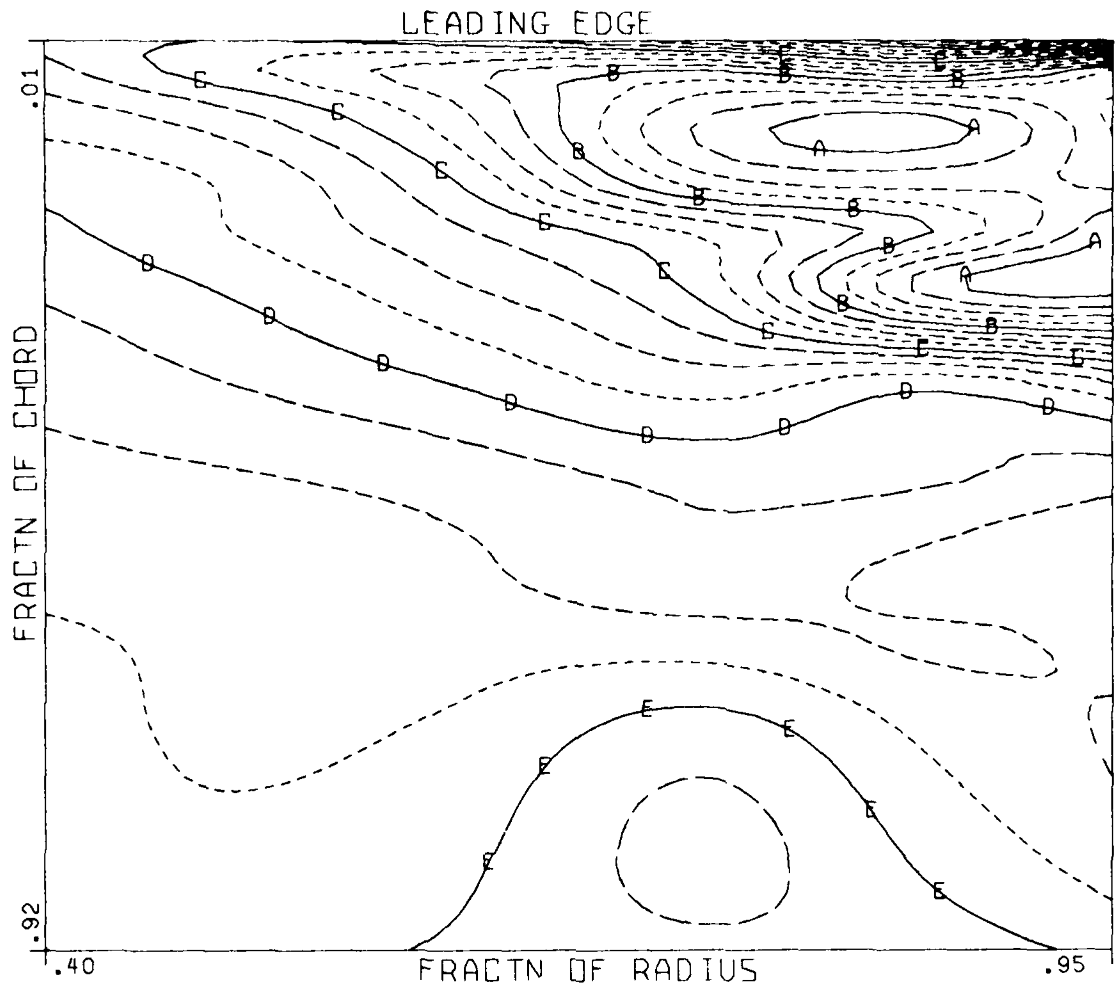
Blade static pressure coefficient, C_p, can be computed from the blade absolute pressure data and airspeed, static pressure, temperature, and azimuth information. Figure 6 shows a multiple curve plot of C_p as a function of chord position for the top blade surface at a specific azimuth position. Figure 7 shows the same data but also shows the bottom surface C_p data in a plotting area with a separate dependent variable scale.

Normal force coefficient, C_n, can be integrated from C_p values. Figures 8 and 9 are contour and surface plots of C_n. Figures 10 and 11 are contour and surface plots of C_n from C81 simulation. Figure 12 is a plot that compares C_n values from OLS and C81 for a particular span station.

1.3 SYSTEM STRUCTURE

DATAMAP consists of two major programs, the File Creation Program and the Processing Program, as well as several utility programs. The File Creation Program reads data from some storage medium (e.g., OLS digital tape or DTF disc file), selectively transfers data to the Master File, and creates a directory of the data stored on the Master File (Figure 13). The Processing Program accepts user commands interactively or in batch mode, retrieves data from the Master File, processes data, and outputs data in graphic or printed format (Figure 14). Thus, the primary product of the File Creation Program, the Master File, is the data input source for the Processing Program.

Four other utility programs are available in the system: the Master File Initialization Program, the Master File Utility Program, the Question and Answer Program to create user input in a directed mode for the File Creation Program, and the Command Sequence File Initialization Program.



LEVEL FLIGHT AT 129 KNOTS   OLS DATA

CYCLE AVERAGE: BLADE ABSOLUTE PRESSURE

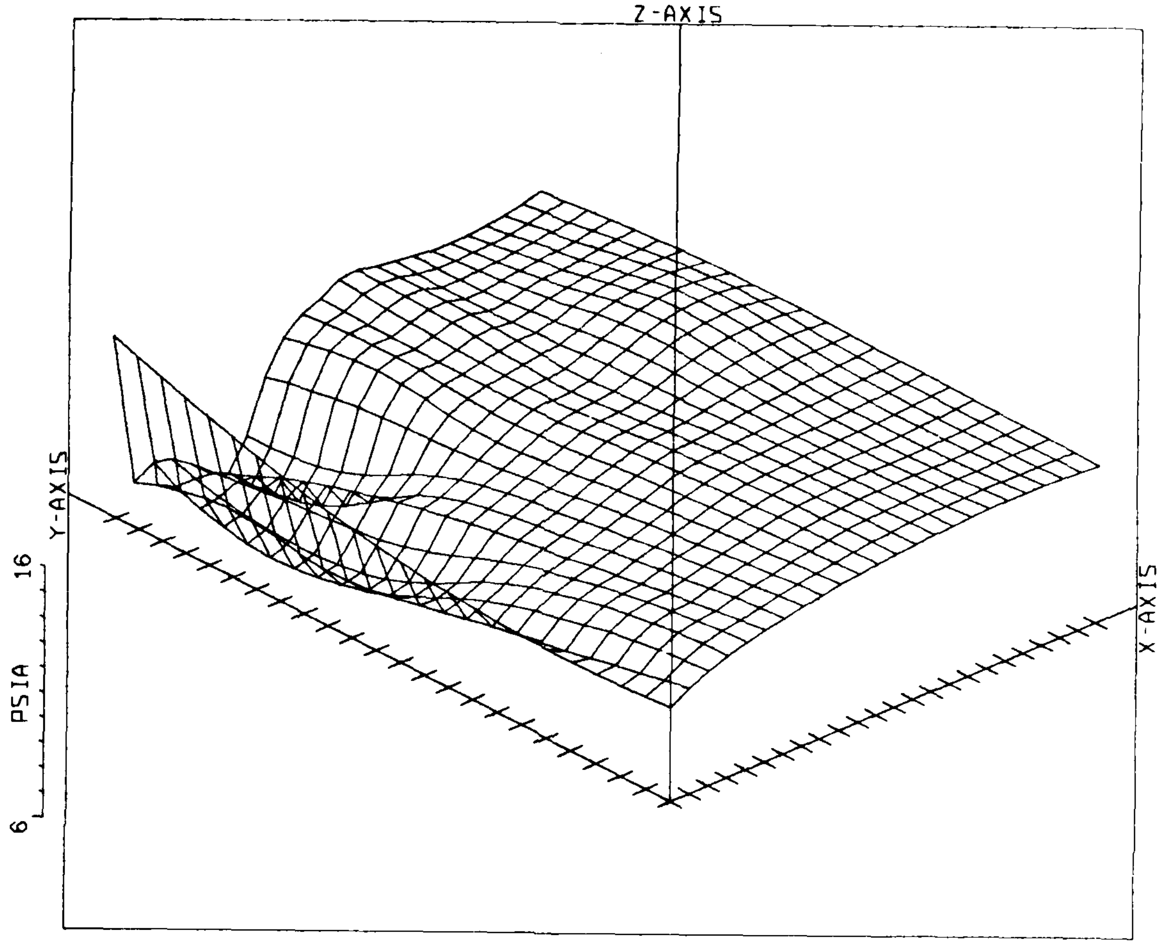
COUNTER	615	GROSS WT	8300	SHIP MODEL	AH-IG
90.00	DEG	LONG CG	200.6	TOP SURFACE	

CONTOUR LEVEL VAUES IN PSIA

A	7.2
B	8.8
C	10.4
D	12.0
E	13.6

BHT.USARTL DATAMAP (VERS 3.00 03/29/80) 04/15/80

Figure 2. Contour plot of blade absolute pressure on top surface of the blade for one azimuth position.



LEVEL FLIGHT AT 129 KNOTS   OLS DATA

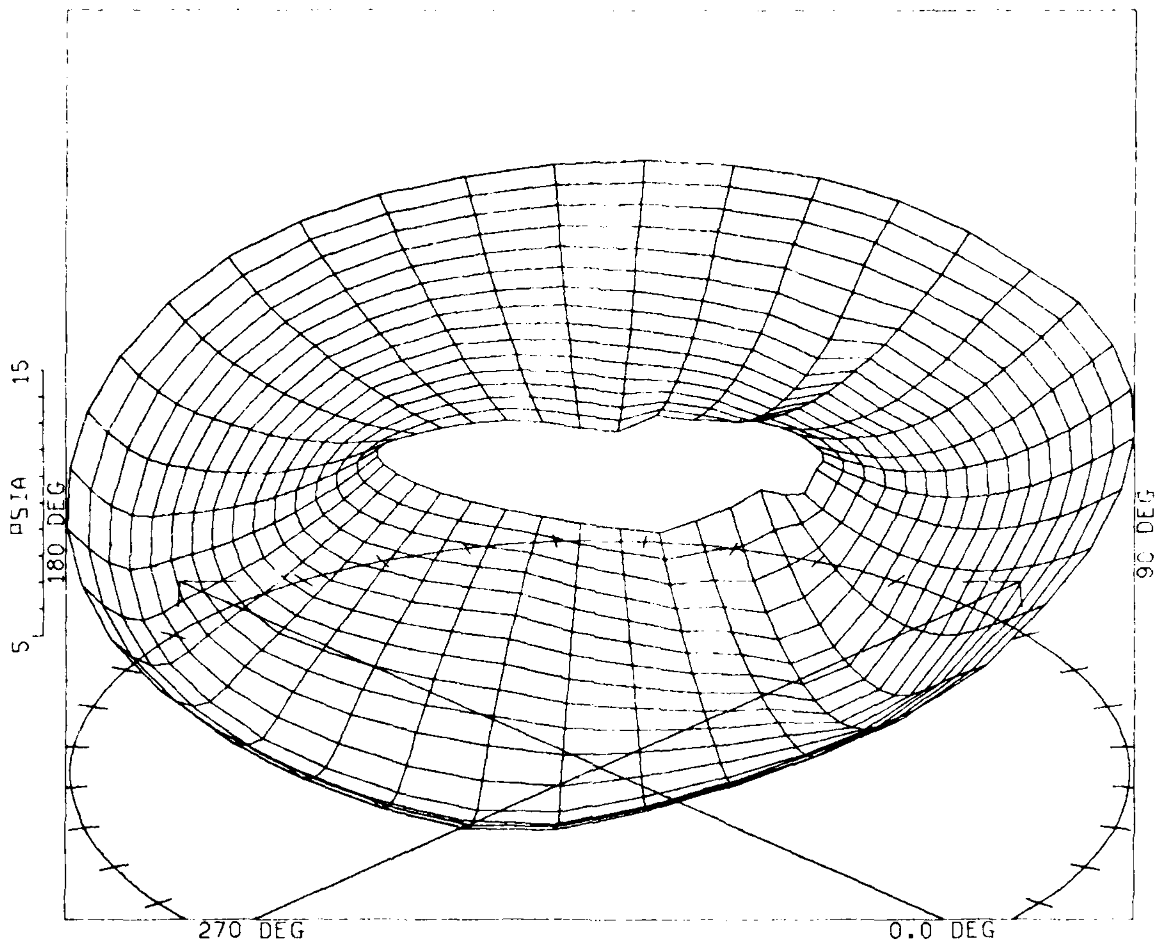
CYCLE AVERAGE: BLADE ABSOLUTE PRESSURE

COUNTER	615	GROSS WT	8300	SHIP MODEL	AH-1G
90.00	DEG	LONG CG	200.6	TOP SURFACE	

X QUANTITY - FRACTN OF CHORD
MIN X .010   MAX X .919   INC X .040
Y QUANTITY - FRACTN OF RADIUS
MIN Y .400   MAX Y .955   INC Y .024
AXES DISPLACED TO MIN RANGE AND DOMAIN VALUES
MIN Z 6.831

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/15/80

Figure 3. Surface plot of blade absolute pressure on the top surface of the blade.



LEVEL FLIGHT AT 129 KNOTS   OLS DATA

CYCLE AVERAGE: BLADE ABSOLUTE PRESSURE

COUNTER 615   GROSS WT 8300   SHIP MODEL AH-1G
.01 X/CHORD   LONG CG 200.6   TOP SURFACE

ANGULAR INCREMENT 10 DEG
RADIAL QUANTITY   FRACTN OF RADIUS
MAX RADIUS .955
RADIAL INCREMENT .0370

BHT,USARTL DATAMAP (VERS 3.50

Figure 4. Cylindrical format surface plot of blade absolute pressure for the leading edge, top surface sensors.

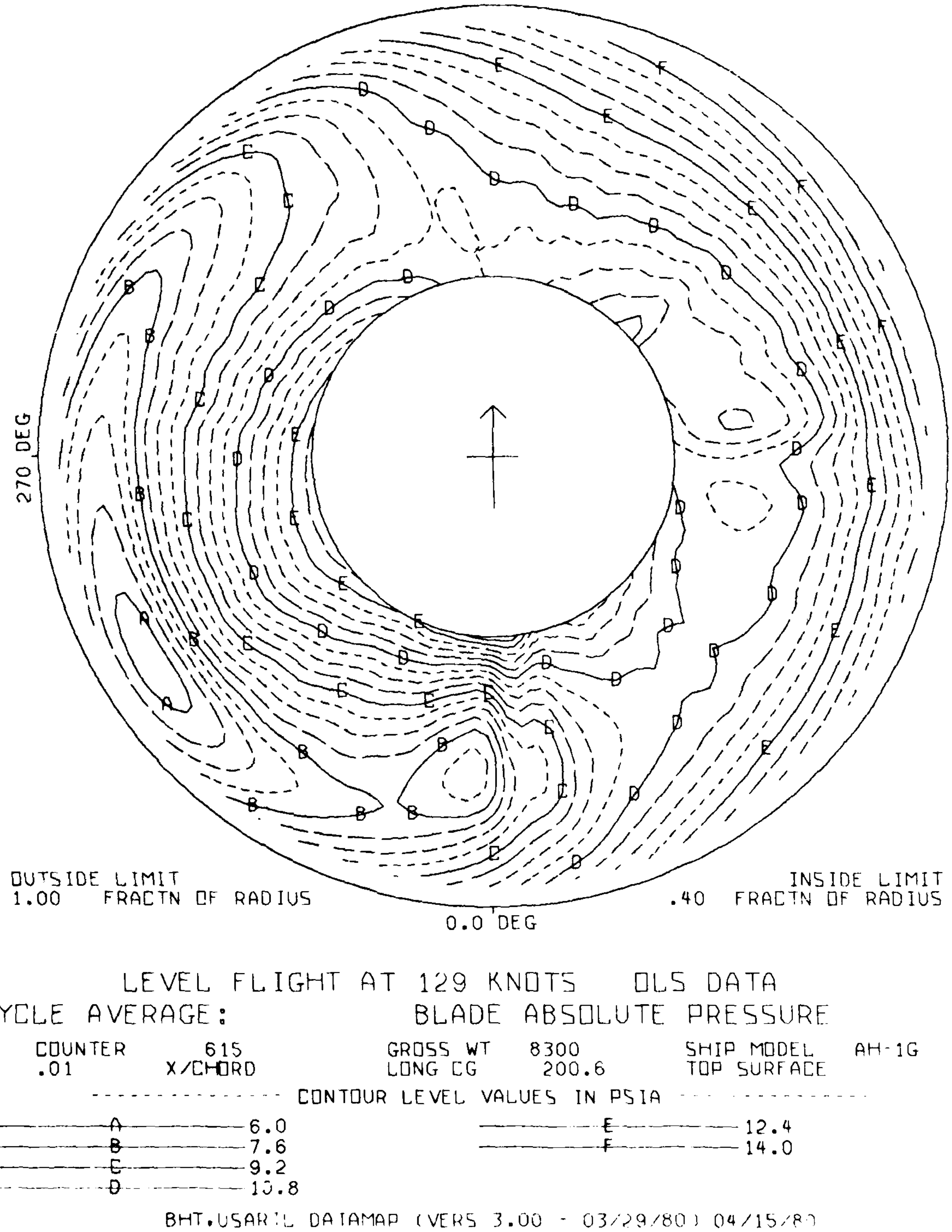
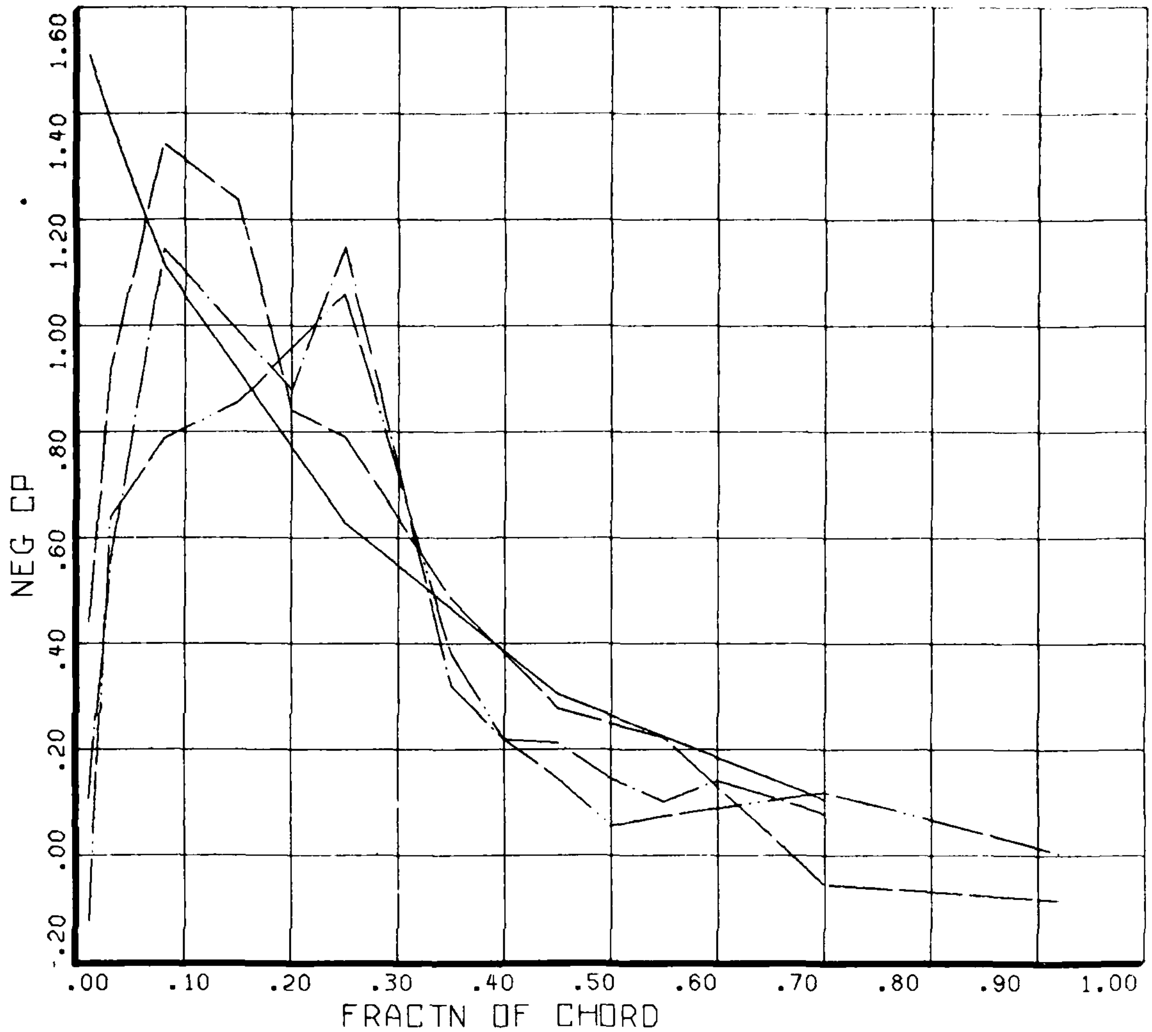


Figure 5. Cylindrical format contour plot of blade absolute pressure for leading-edge, top-surface sensors.



LEVEL FLIGHT AT 129 KNOTS OLS DATA

DERIVED PARAMETER: BLADE STATIC PRESSURE COEFF

COUNTER 615 GROSS WT 8300 SHIP MODEL AH-1G
90.00 DEG LONG CG 200.6 UPPER SURFACE

.40 R/RADIUS
.75 R/RADIUS
.86 R/RADIUS
.95 R/RADIUS

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/16/80

Figure 6. Cycle-averaged blade static pressure coefficient on the top surface of the blade for one azimuth position.

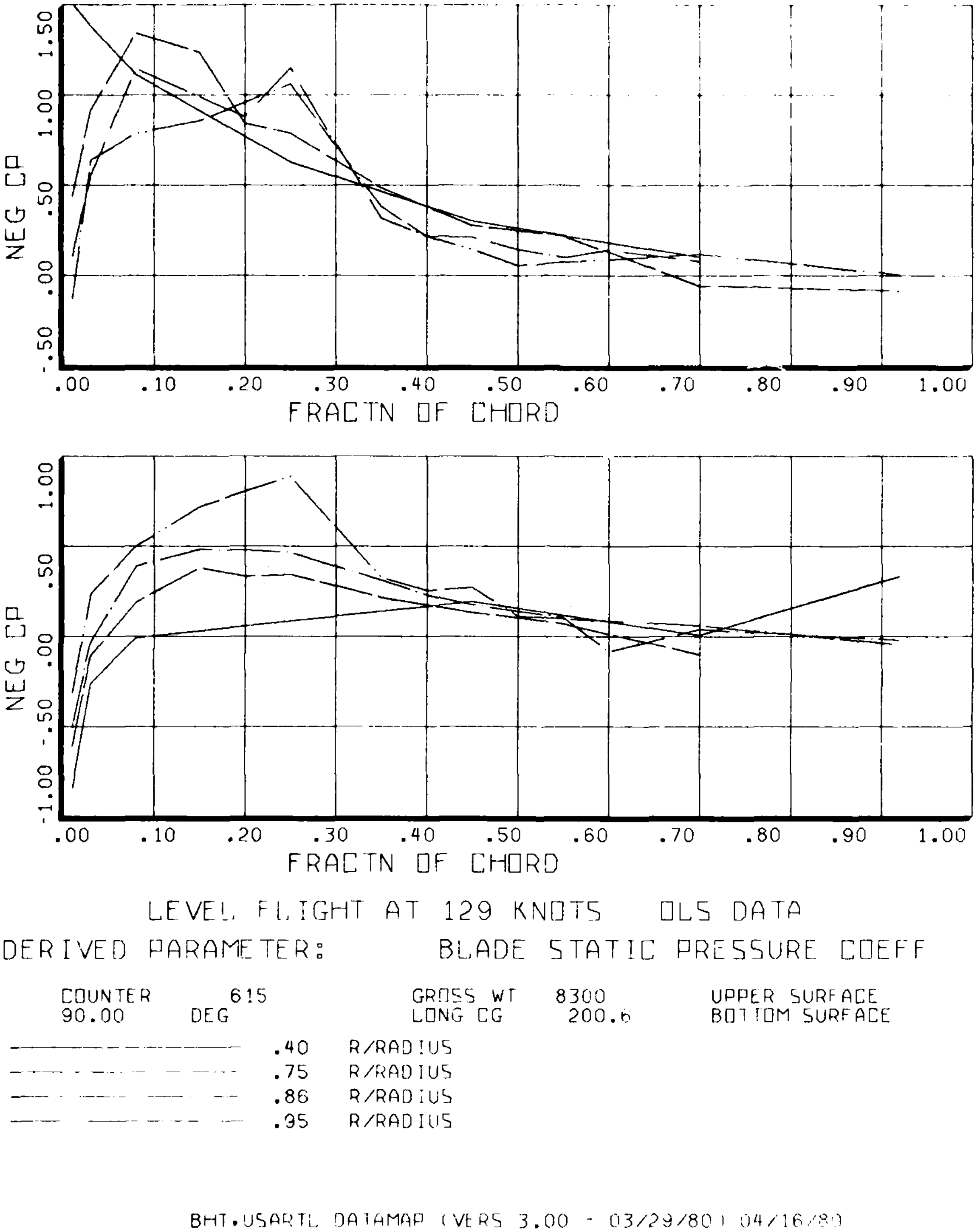


Figure 7. Cycle-averaged blade static pressure coefficient on the top and bottom surfaces of the blade for one azimuth position.

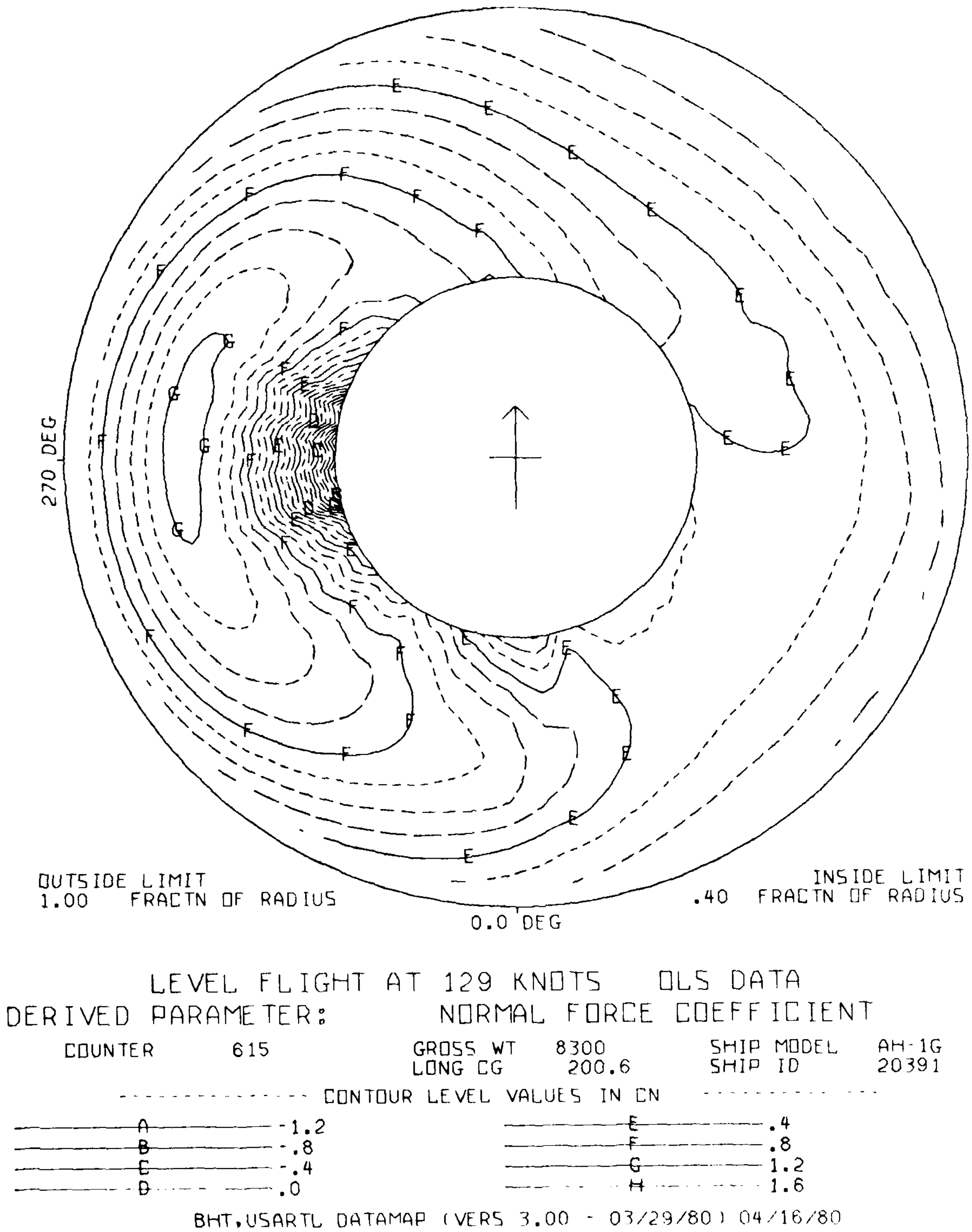
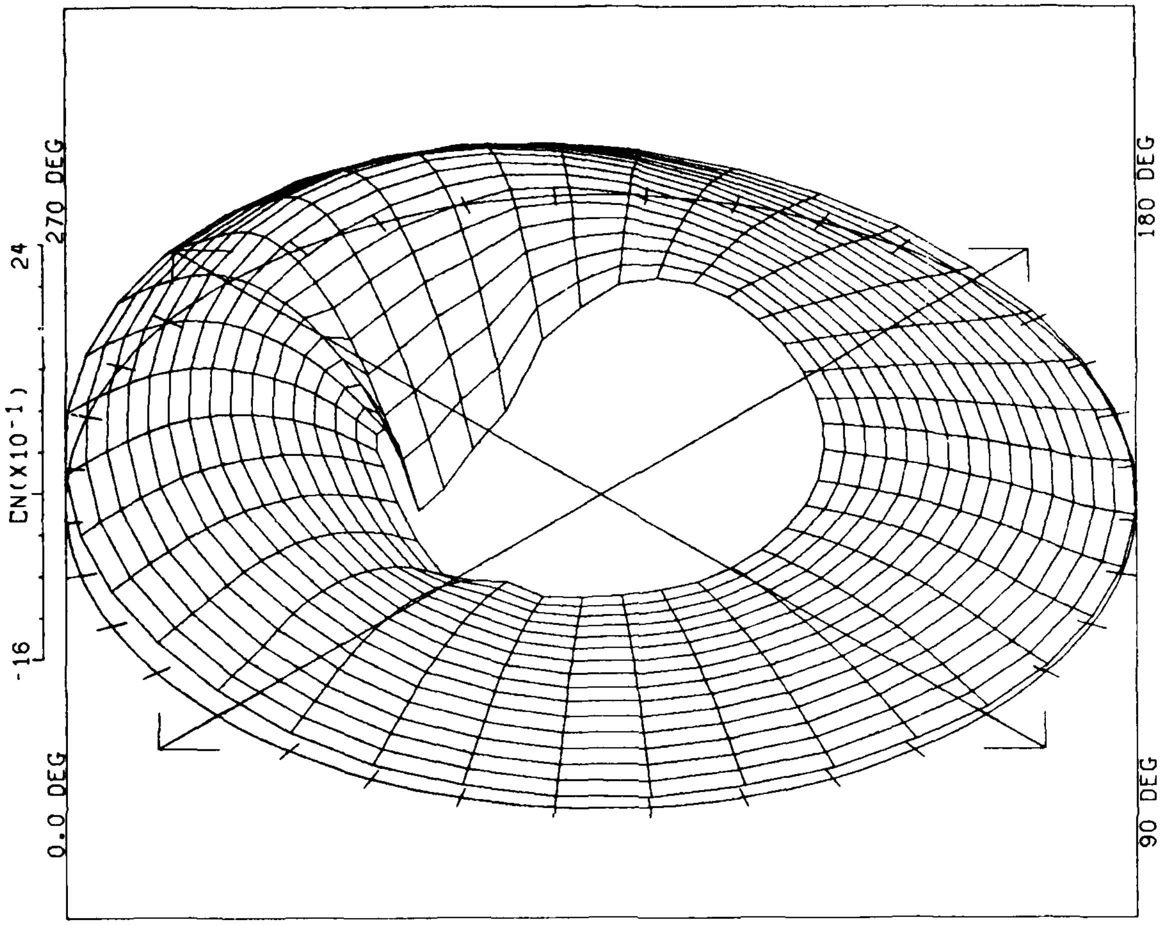


Figure 8. Cylindrical format contour plot of normal force coefficient.



LEVEL FLIGHT AT 129 KNOTS   OLS DATA

DERIVED PARAMETER: NORMAL FORCE COEFFICIENT

COUNTER   615   GROSS WT   8300   SHIP MODEL   AH-1G
LONG CG   200.6   SHIP ID   20391

ANGULAR INCREMENT 10 DEG
RADIAL QUANTITY   FRACTN OF RADIUS
MAX RADIUS   .955
RADIAL INCREMENT   .0370

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/16/80

Figure 9. Cylindrical format surface plot of normal force coefficient for all azimuth and radial positions.

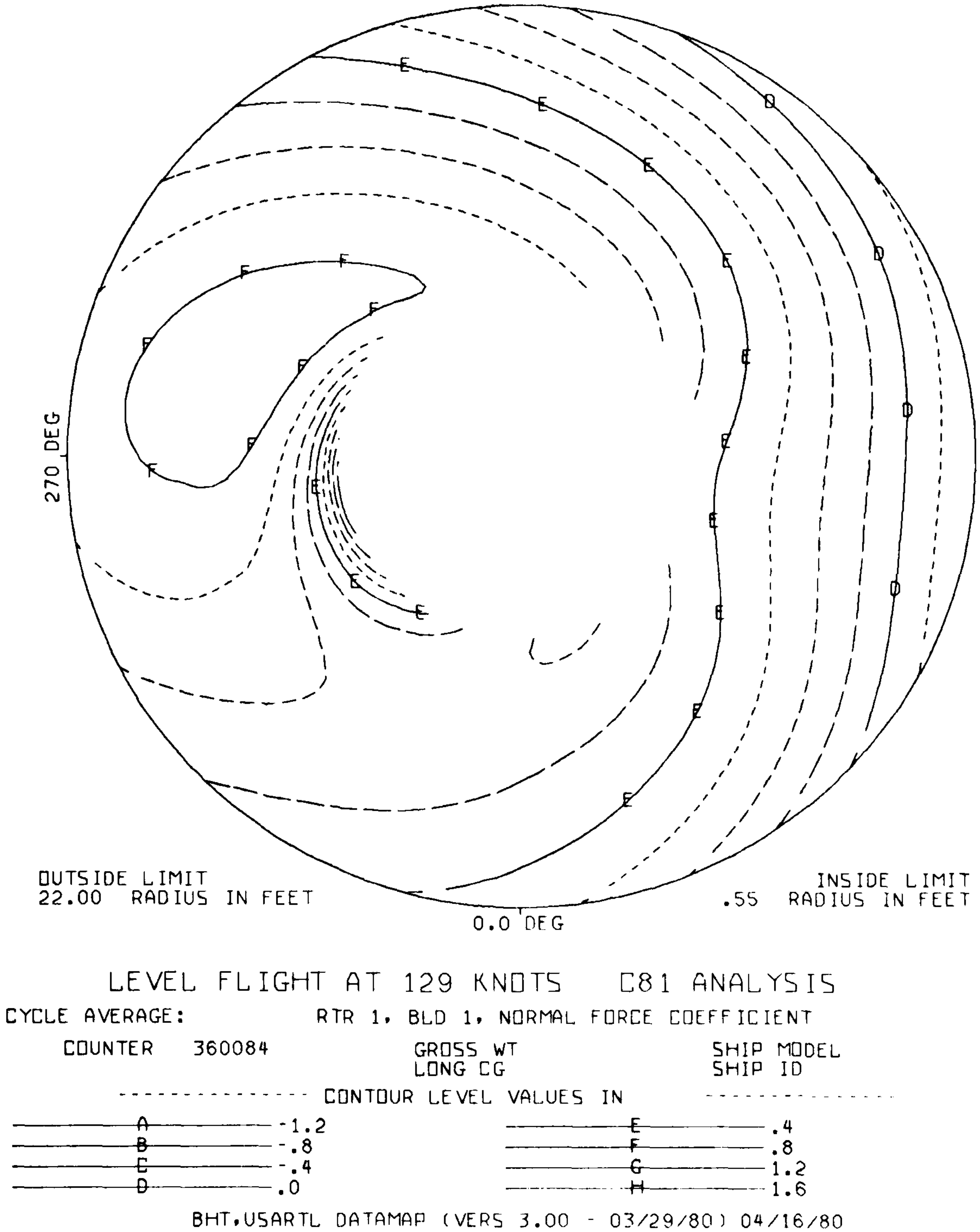
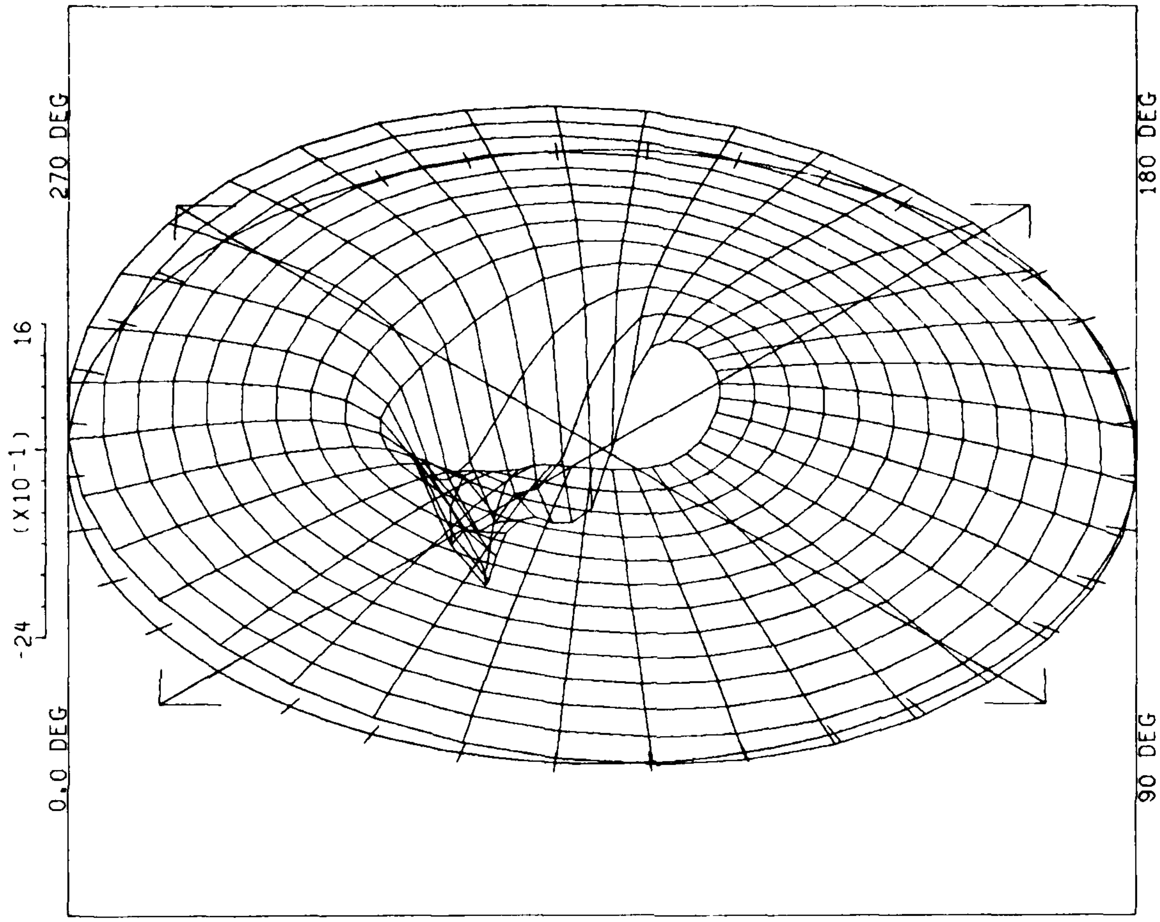


Figure 10. Cylindrical format contour plot of C81-generated normal force coefficient for all azimuth and radial positions.



LEVEL FLIGHT AT 129 KNOTS   C81 ANALYSIS

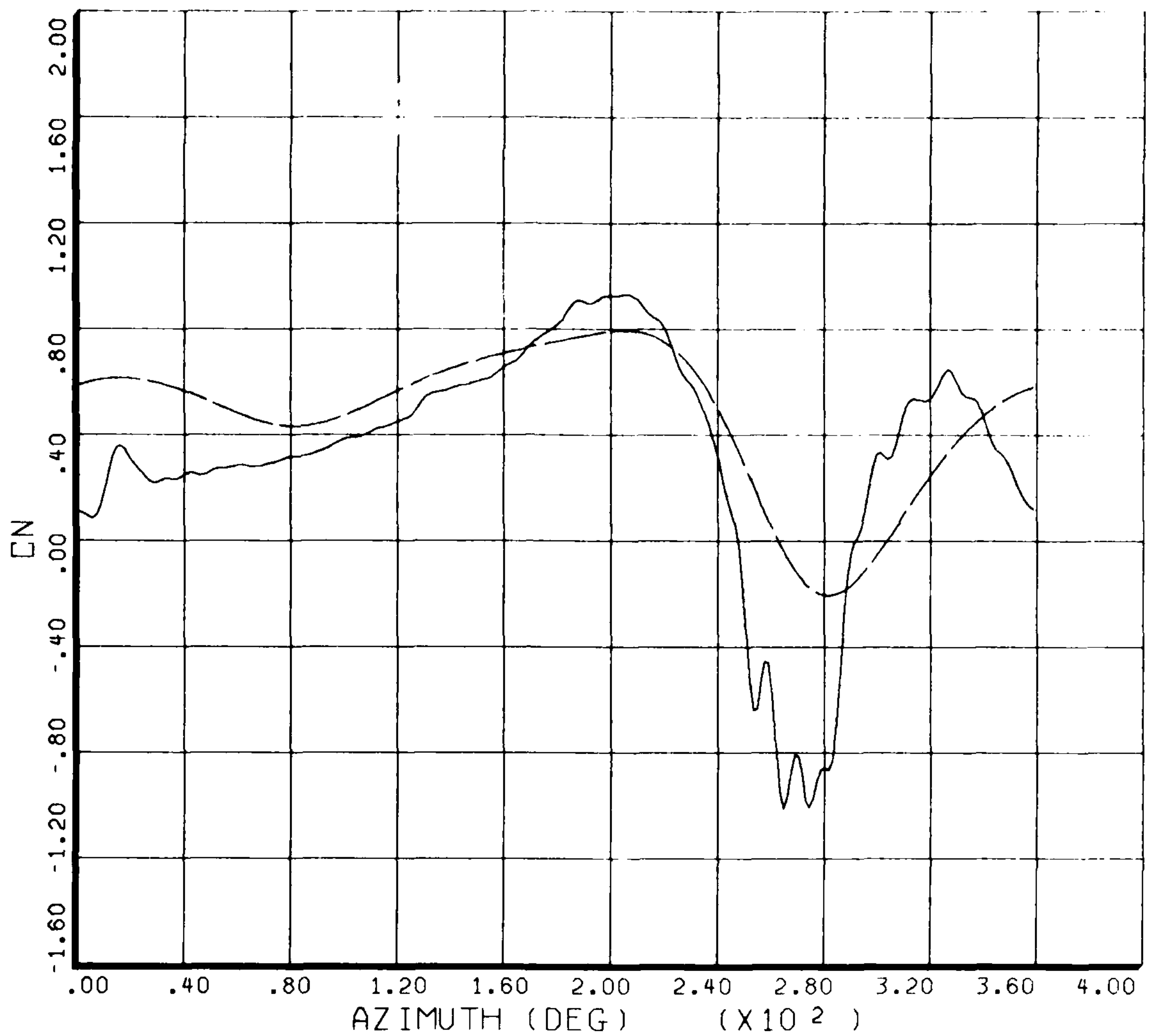
CYCLE AVERAGE: RTR 1, BLD 1, NORMAL FORCE COEFFICIENT

COUNTER 360084   GROSS WT   SHIP MODEL
LONG CG   SHIP ID

ANGULAR INCREMENT 10 DEG
RADIAL QUANTITY   RADIUS IN FEET
MAX RADIUS 22.000
RADIAL INCREMENT 1.4300

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/16/80

Figure 11. Cylindrical format surface plot of C81-generated normal force coefficient for all azimuth and radial positions.



______________ COUNTER	615	GROSS WT 8300	SHIP MODEL	AH-1G
		LONG CG 200.6	SHIP ID	20391
OLS DATA FOR 129 KNOTS	40 PERCENT SPAN			
DERIVED PARAMETER:	NORMAL FORCE COEFFICIENT			
____ ____ ____ COUNTER	360084	GROSS WT	SHIP MODEL	
		LONG CG	SHIP ID	
C81 ANALYSIS FOR 129 KNOTS	39 PERCENT SPAN			
CYCLE AVERAGE:	RTR 1, BLD 1, NORMAL FORCE COEFFICIENT			

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/16/80

Figure 12. Direct comparison of C_n from OLS and C81 for a particular span station.

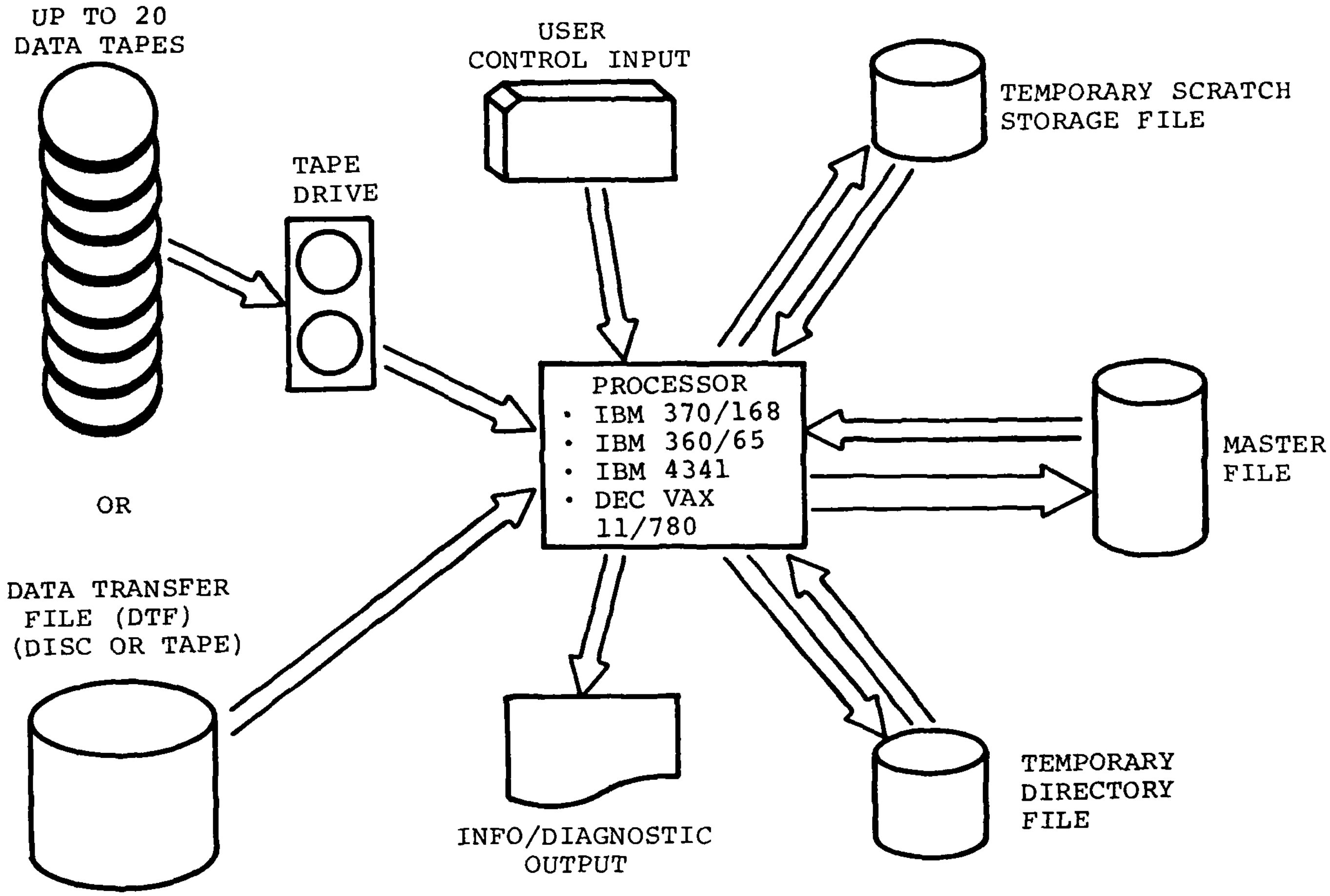


Figure 13. Information flow for File Creation Program.

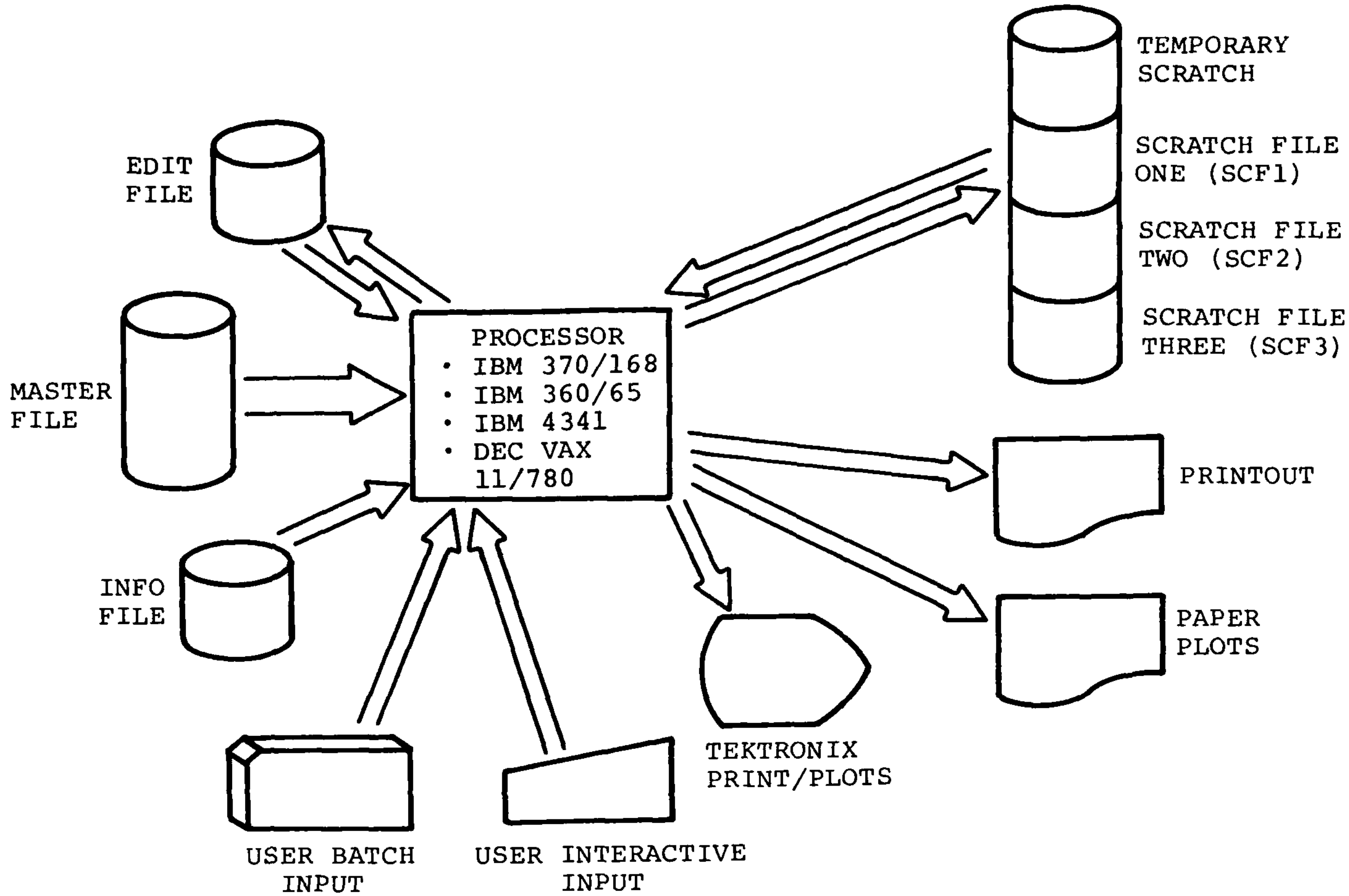


Figure 14. Information flow for Processing Program.

1.4 SCOPE OF MANUAL

The purpose of Volume I of this report is to delineate the general structure and capabilities of DATAMAP and to explain the commands and inputs provided to use this package. Mathematical techniques used in the analyses and derivations will also be explained. Volume II of this report provides information on programming considerations. That is, program flow, overlay structure, disc file definitions, computer system dependencies, and graphics software interfaces are described. In addition, Job Control Language (JCL) required to compile, link, and execute the various programs that comprise DATAMAP is described. Examples of JCL and IBM Time Sharing Option (TSO) language used on an IBM 370/168 MVS system are given.

1.5 OVERVIEW OF MANUAL

Section 2 of this volume describes the capabilities of the various programs that comprise DATAMAP and introduces many concepts and terms. Section 3 explains the procedures required to initialize the Master File. In addition, procedures and considerations for placing data on the Master File are described together with maintenance functions.

The last three sections deal with the Processing Program. Section 4 introduces the syntax for user input, Section 5 covers the available command entry sequences, and Section 6 discusses each of the analysis and derivation methods used in the program.

DATAMAP was written to be as general in application as possible and the DTF format for input to DATAMAP is intended to provide access to many test and simulation data bases. However, in this report explanations of procedures and capabilities are often illustrated with examples from the OLS data base and/or C81 simulations. These examples should not be taken to indicate that a specific OLS or C81 example is the only available application of a system capability.

2. SYSTEM PROCESSING CAPABILITIES

2.1 DATA STORAGE

DATAMAP stores measured and simulated data on the Master File in direct access format. The storage space on the Master File is distributed into partitions of contiguous records and unused space. DATAMAP creates a partition when it writes data onto the file, or it can add data to an existing partition. A user will wish to store a body of related data on one partition for simultaneous processing. Directories in the Master File and in the partitions allow immediate access to stored data without reading unnecessary data records.

Conceptually, the system was designed to provide a separate partition for each user, although several users can use a single partition or one user may use several partitions. When viewed over an extended period of time, multiple, variable-size partitions provide access for several users to a relatively large amount of disc space without requiring the assignment of an exceedingly large amount of total disc space. Each user may add data to the Master File as required and, similarly, each user may delete data from the file as the need arises. However, the data contained in each partition are protected by a password that is assigned by the user when the partition is defined. Each installation should have one person assigned to monitor the Master File and to assure that unnecessary data are deleted. The Master File Utility Program, as described in Paragraph 3.4, is available to support this function.

Data are stored on the Master File as a sequence of discrete values that are indexed by time. These data sequences are often referred to as time histories. Each time history is referenced by item code and counter. The item code is a string of 4 characters that uniquely identifies the output from a particular sensor. A counter is a number between 1 and 999,999 that uniquely identifies a single maneuver. More generally, a counter references a period of time when useful data were taken or a simulation of a specific maneuver or flight condition. An item code/counter pair may appear only once inside any partition, although it may appear in several different partitions.

There is no specific limit to the number of item codes or counters that can appear in a partition. Similarly, there is no specific limit to the number of data samples that comprise a time history. However, the Master File data set and the partition must be sufficiently large to hold the data.

OLS data are recorded on digital tape in an uncalibrated, integer form and each integer value may be represented using 16 bits. Calibration constants are available on tape to provide conversion to engineering units. However, calibrated data values require 32 bits for computer representation. Data can be stored on the Master File in either the integer or calibrated form. Integer format data require as little as half the space required to store calibrated data. In order to retain as much accuracy as possible, the data are always stored in a calibrated form whenever digital filters are applied before storage. Data stored in DTF format are always in calibrated form.

Each time history stored on the Master File is preceded by an information record that contains useful information about the data stream. Information stored in the information record includes calibration factors, description of the data, units, sampling rate, amount of data present, and other parameters.

2.2 DATA TRANSFER TO STORAGE

As mentioned previously, the File Creation Program will accept data for transfer to the Master File in either BHT-Ground Data Center (GDC) or Data Transfer File (DTF) format. GDC format data are usually read from one or more digital tapes. The storage medium for a DTF may be tape or disc.

2.2.1 Objectives for DTF Usage

Figure 15 shows the basic data paths for the specific cases of C81 and GDC format data. C81 passes output data through intermediate disc files to the C81 Postprocessor, which in turn writes the DTF. The File Creation Program (FCP) reads either the DTF or GDC format tapes and transfers the data to the Master File. However, the DTF format will allow a more comprehensive class of data bases to be interfaced to DATAMAP as depicted in Figure 16. First, output from existing simulation programs could be run through a format conversion program to create a DTF. The format conversion program would correspond to the C81 Postprocessor in Figure 15. Second, a format conversion program might also be written to convert non-GDC format test data to a DTF. Finally, a new simulation program could be written or an existing program could be converted to write data in DTF format directly as a primary output.

Figure 17 shows an additional consideration for data paths. A simulation could execute in one type of computer and write a DTF. This DTF might then be read by the FCP while running on a different type of computer. Similarly, a conversion program running on one computer might be used to reformat test data

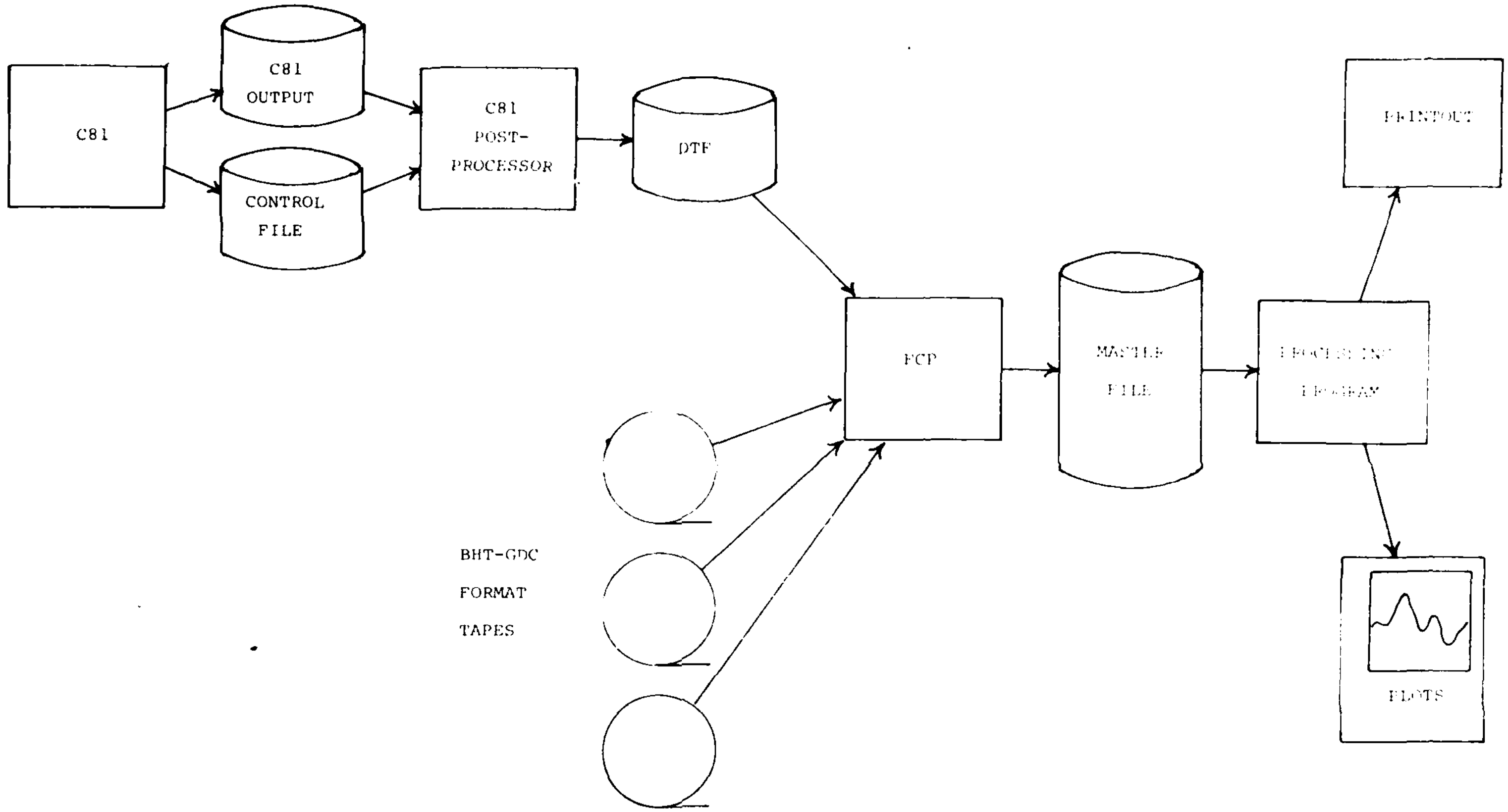


Figure 15. Information flow for C81-generated and GDC-format data.

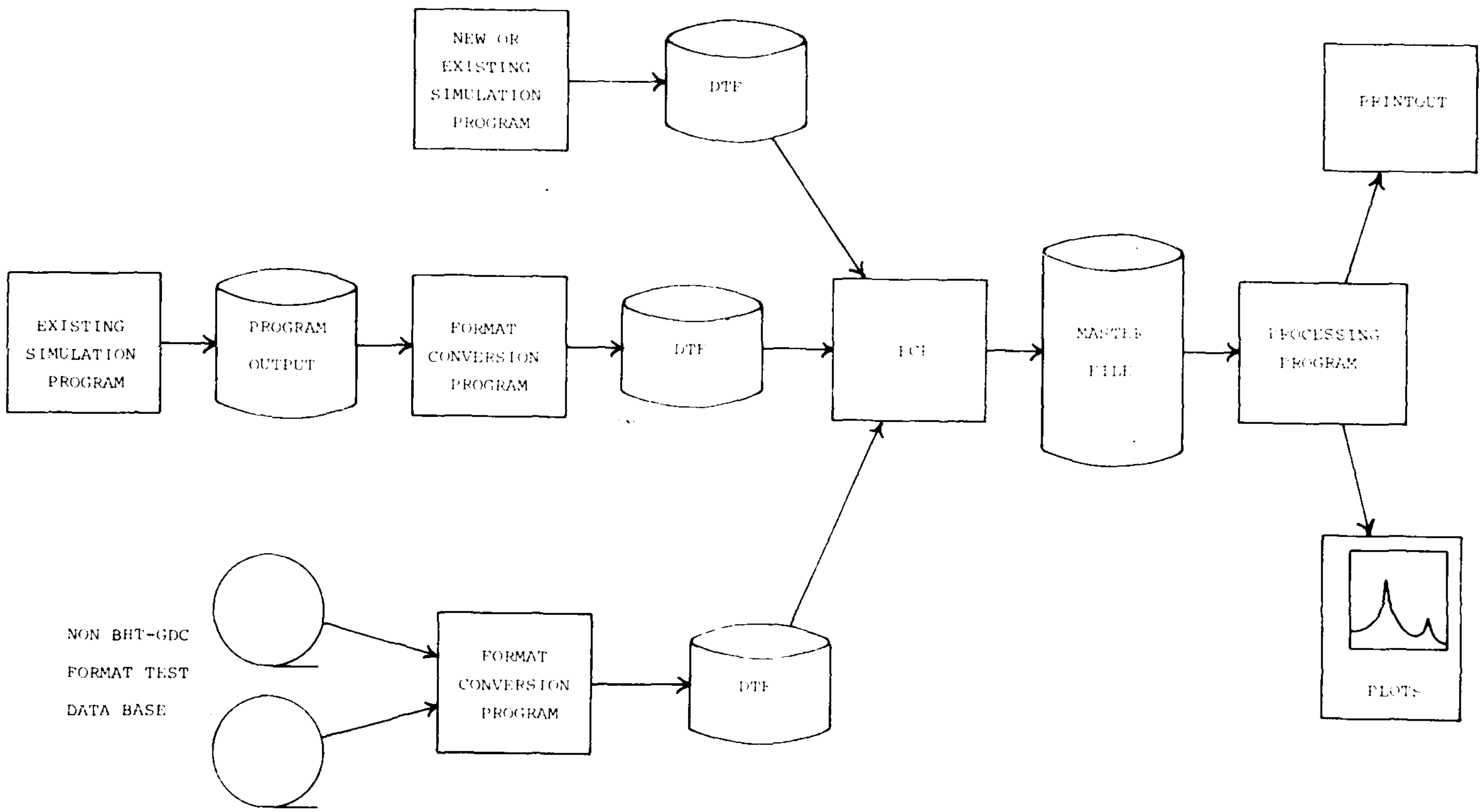


Figure 16. Information flow for other data bases into DATAMAP.

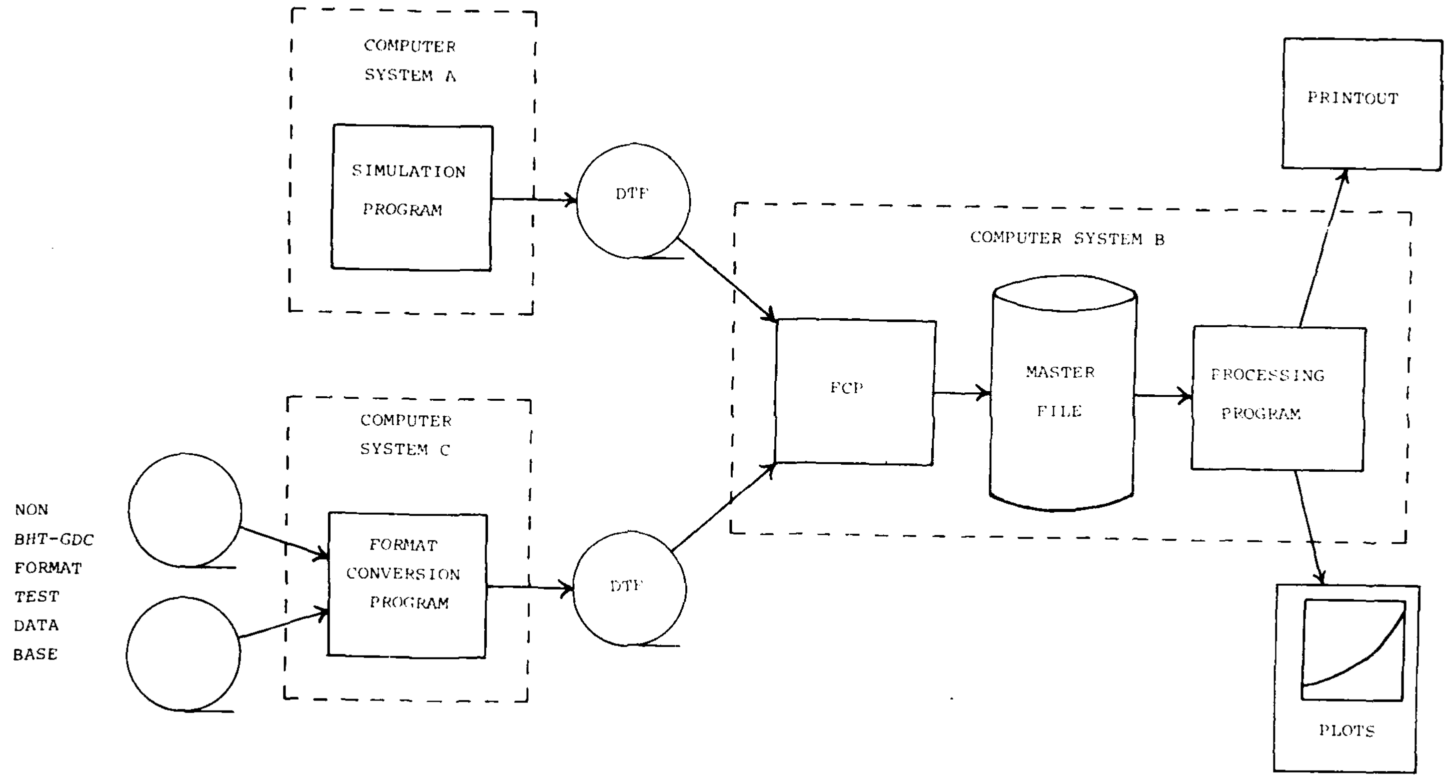


Figure 17. Information flow for transfer of data between different computers.

into a DTF and then the DTF could be read by the FCP running on another system. The DTF format will make such a data transfer between different computers reasonably convenient. Notice that the DTF is depicted as a tape in Figure 17 although some computer configurations would allow a disc to be used for transfer of data between computers.

2.2.2 File Creation Program Processing and Selection Options

As depicted in Figure 13, user instructions are read by the FCP. These instructions are read from the system input file (e.g., the card reader) or they may be included in the DTF. The purpose of the input is to select the data that will be transferred to the Master File and to specify any processing that will be applied to the data during the transfer process. The user may select the counters and item codes to be transferred and, for the time histories specified, he may select a restricted segment of the histories for transfer. Processing options include low-pass digital filtering and modification of time history sample rates. For OLS data, alignment processing is required, and for all GDC format data, calibration of the data is possible. For DTF input, considerably more latitude is available in modification of sample rate as a new sample rate may be specified arbitrarily. Sample rates for data stored in BHT-GDC format may only be reduced by integer factors. Consult Paragraph 3.2.3 for considerations in modification of sample rates and use of digital filters.

2.3 DATA RETREIVAL

At the beginning of execution of the Processing Program, the user specifies the partition of the Master File to be used. Later, during the Processing Program run, the user may change the partition that is accessed and/or two partitions may be accessed simultaneously. Data are retrieved from the Master File as needed by reference to item code, counter, offset from the beginning of data present, and amount of data wanted. This information is specified by the user either directly or implicitly. Data stored in integer format are calibrated and when the data are retrieved, an information record for the time history is also retrieved from the Master File for the program to use in generating various plot or printout labels. The information record alone may be retrieved to generate a display for the user of information about item codes present for a given counter in the partition.

Any number of Processing Program users may access data from one or more partitions of the Master File at the same time. However, if one job is accessing the file with the File Creation Program, no other job or users may access the Master File at that time.

2.4 ANALYSES

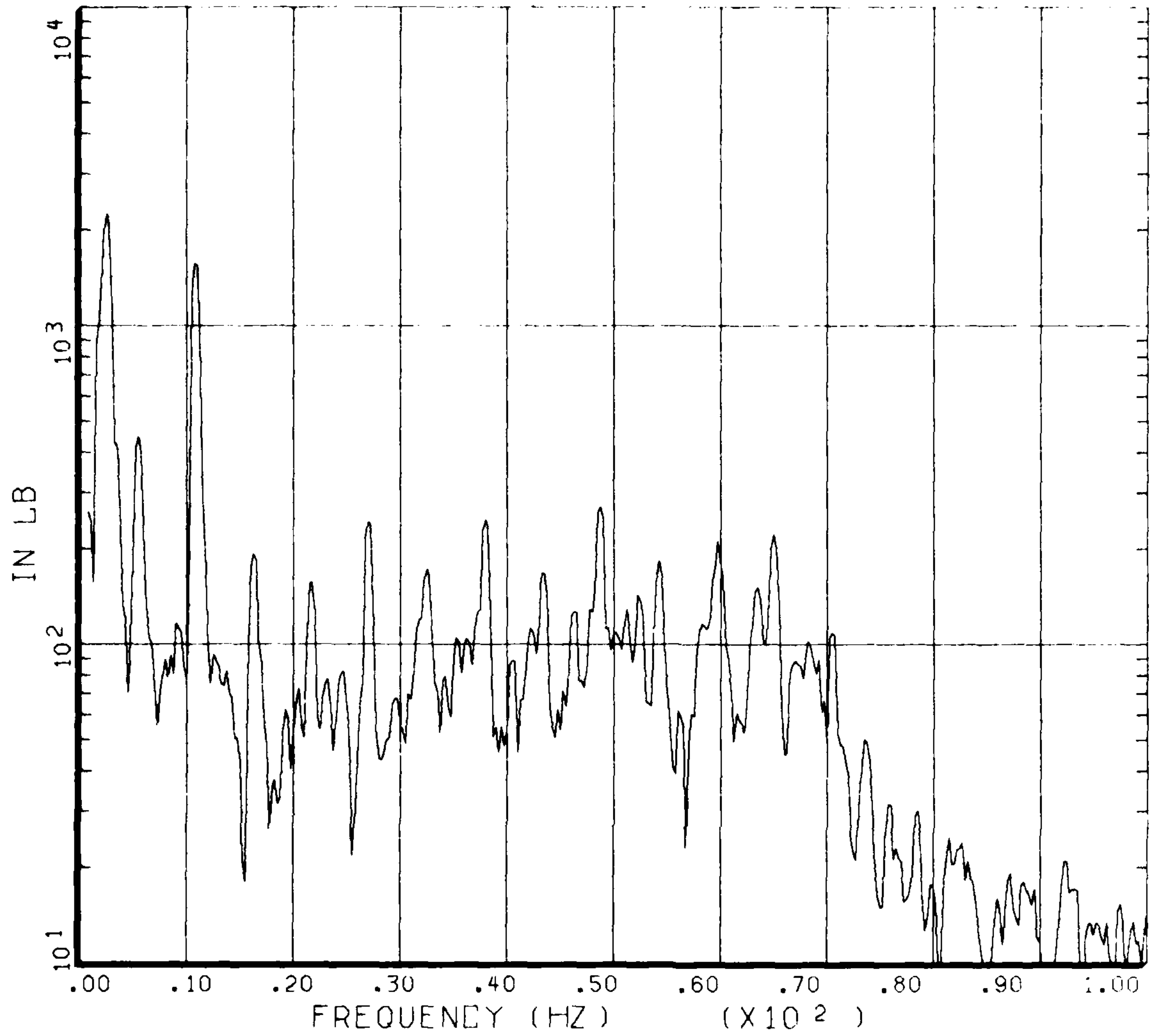
Following is a brief description of each of the analyses that are available in the DATAMAP Processing Program. Descriptions of the analysis output functions and processing options for each analysis are given as well. Section 6.1 describes the method of computation of each analysis function.

The Amplitude Spectrum option uses a Fast Fourier Transform (FFT) algorithm to represent functions of time as functions of frequency. Displays or plots show amplitude as a function of frequency (Figure 18). Input functions of time consisting of well separated, relatively pure frequency components yield amplitude spectra that have been optimized to accurately predict amplitude of the various frequency components. Rectangular, Cosine Taper, or Hanning Window functions can be specified for use in the Amplitude Spectrum option. The beginning user is encouraged to specify the Cosine Taper (COS) window. Windows are discussed in more detail in Section 6.1.

Cycle-Averaging takes a time history consisting of two or more complete rotor cycles along with a table of the end points of each cycle and averages together the cycles to obtain one representative cycle. The operation tends to reduce superfluous noise and reduces the effect of unique deviations in single rotor cycles. Cycle-Averaging is also a useful method to obtain a single rotor cycle of data for display or processing. In particular, cylindrical format contour and surface plots require complete rotor cycles for input, and Cycle-Averaging is often the easiest way to define the cycle. In addition, the Cycle-Averaging process interpolates to obtain a record of 256 values, which is a power of two. This feature can simplify analyses that use the Fast Fourier Transform (FFT) algorithm.

Harmonic Analysis computes the Fourier Series components for a user-specified integral number of rotor cycles that correspond to an integer multiple of the number of cycles. Output is amplitude and phase for each component versus harmonic number or frequency (Figure 19). The user may specify the number of harmonics calculated and, if the number of harmonics is specified as one, the user may specify the particular harmonic to calculate.

Digital Filtering will low-pass or band-pass filter a time history to eliminate undesirable frequency components. This filtering capability is separate from the low-pass filtering capability in the FCP. The user specifies the upper and lower break-frequencies and the number of poles in the mathematical



800 FPM DESCENT AT 37 KNOTS

AMPLITUDE SPECTRUM: MR MAST TORQUE

COUNTER	956	GROSS WT	8300.	SHIP MODEL	[illegible]
		LONG CG	196.3	SHIP ID	[illegible]

956/M107

BHT.USARTL DATAMAP (VERS 3.00 [illegible]

Figure 18. Amplitude spectrum plot (semi-log scale).

HARMONIC ANALYSIS: MR MAST TORQUE PAGE 1

COUNTER 956

HARMONIC NUMBER	AMPLITUDE IN LB	PHASE DEGREES
0.0	0.429147E+05	0.0
0.100000E+01	0.453264E+03	-0.373145E+00
0.200000E+01	0.160564E+04	0.848373E+01
0.300000E+01	0.148881E+03	-0.137304E+03
0.400000E+01	0.180008E+03	0.503054E+02
0.500000E+01	0.245368E+03	-0.168055E+03
0.600000E+01	0.134244E+03	0.460350E+02
0.700000E+01	0.237408E+03	-0.179177E+03
0.800000E+01	0.743447E+02	0.651059E+02
0.900000E+01	0.302646E+03	-0.119482E+03
0.100000E+02	0.996052E+02	0.436956E+02
0.110000E+02	0.201839E+03	-0.999669E+02
0.120000E+02	0.735042E+02	0.991963E+02
0.130000E+02	0.558947E+02	-0.444476E+02
0.140000E+02	0.275244E+02	0.418989E+02
0.150000E+02	0.231400E+02	-0.394527E+02
0.160000E+02	0.801179E+01	0.736782E+02
0.170000E+02	0.532265E+01	-0.102167E+03
0.180000E+02	0.119389E+02	0.118652E+03
0.190000E+02	0.753928E+01	-0.179713E+03
0.200000E+02	0.110467E+02	0.982673E+02

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/19/80

NEW STEP.

Figure 19. Harmonic Analysis printout.

representation of the filter. If the lower break-frequency is zero, the filter becomes a low-pass filter. Frequency components of the time history between the breakpoints in the pass band will be essentially unchanged in magnitude and phase. Frequency components outside the pass band will be greatly reduced in magnitude. Generally, more attenuation can be achieved outside the pass band by increasing the number of poles in the mathematical representation of the filter although three poles are usually sufficient. Filter users should be aware that the end point areas of time histories may be distorted by digital filtering. Usually, the problem is magnified by increasing the number of poles. A special filtering algorithm is used for cycle-averaged input records that should eliminate edge effects unless the upper break-frequency is less than the rotor frequency (see Section 6.1.2).

<u>Moving Block Damping</u> analysis assumes an input function of the form

$$F(t) = Ae^{-\frac{Dwt}{100}}\sin(wt + \phi) + Q(t),$$

where D is percent of critical damping, w is a known or suspected frequency component, and Q(t) is a function with frequency components well separated from w.

The output of moving block damping for a time history is the percent of critical damping, D, where a positive value implies the frequency component is stable.

<u>Min/Max Analysis</u> takes a time history consisting of one or more complete rotor cycles along with a table of the end points of each cycle and finds the maximum and minimum value occurring in each cycle. Oscillatory and mean values are calculated for each cycle where

$$\text{Oscillatory} = 1/2\ (\text{max} - \text{min})$$

$$\text{Mean} = 1/2\ (\text{max} + \text{min})$$

<u>Auto-Spectral Density</u> is sometimes referred to as "power spectral density" or a "power spectrum." Auto-Spectral Density uses an FFT algorithm to estimate the density of variance introduced by each frequency component of a function. Thus, the integral of auto-spectral density is the variance of the original input function. Alternatively, Auto-Spectral Density may be interpreted as the Fourier Transform of the auto-correlation of an input function. Rectangular, Cosine Taper, Hanning, or Half Cosine Window

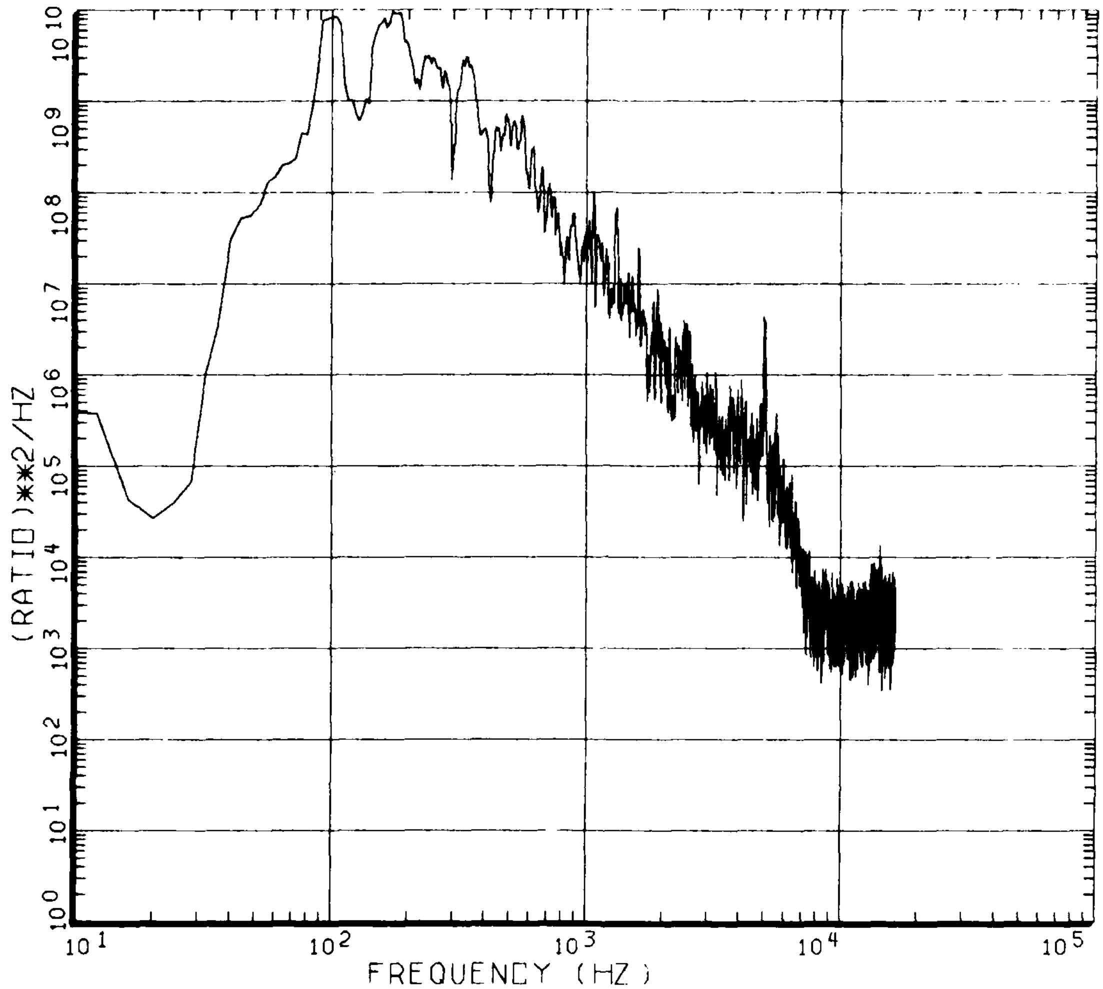
functions can be specified for use in the estimation of auto-spectral density. The beginning user is encouraged to specify the Half Cosine Window for this application. Windows are discussed in more detail in Section 6.1. Ensemble averaging and adjacent point averaging in frequency space are also available for this process. Figure 20 depicts the Auto-Spectral Density for some acoustic data.

Cross-Spectral Density is sometimes referred to as a "cross-power spectrum." Cross-Spectral Density is interpreted as the Fourier Transform of the cross-correlation of two functions. The output of Cross-Spectral Density is complex so that both magnitude and phase output functions of frequency are produced. The same window selection is available for Cross-Spectral Density as for Auto-Spectral Density together with ensemble averaging and adjacent point averaging.

Frequency Response Function is an estimate of the Fourier Transform of the impulse response of a linear system. The estimate is made for one or more pairs of records for which the first record in a pair is assumed to be the input to a linear system and the second is assumed to be the corresponding output from the same system. Both magnitude and phase functions of frequency are produced as output as shown in Figure 21. The same window selection is available for Frequency Response as for Auto-Spectral Density together with ensemble averaging and adjacent point averaging.

Coherence Function is a useful test for the hypothesis that a set of time history pairs represent the input and corresponding output functions for a linear system. Coherence function is a function of frequency with values close to one indicating high coherence (or correlation) and values closer to zero indicating low coherence. The output is a function of frequency, and high coherence may be apparent for some frequency bands and low coherence for other bands. The same window selection is available for the coherence function as for Auto-Spectral Density together with adjacent point averaging. Ensemble averaging is required for this process.

Auto-Correlation is actually calculated and labeled on output as auto-covariance, which is the auto-correlation of a time-history with the mean level reset to zero. Auto-correlation is the integral of the product of a function with the same function offset by a specific time interval. Thus, the Auto-Correlation is a function of this offset. Since Auto-Correlation is an even function, it is displayed only for positive offset values. No window options or adjacent point averaging are provided for Auto-Correlation, but ensemble averaging can be used. A special algorithm is used automatically for



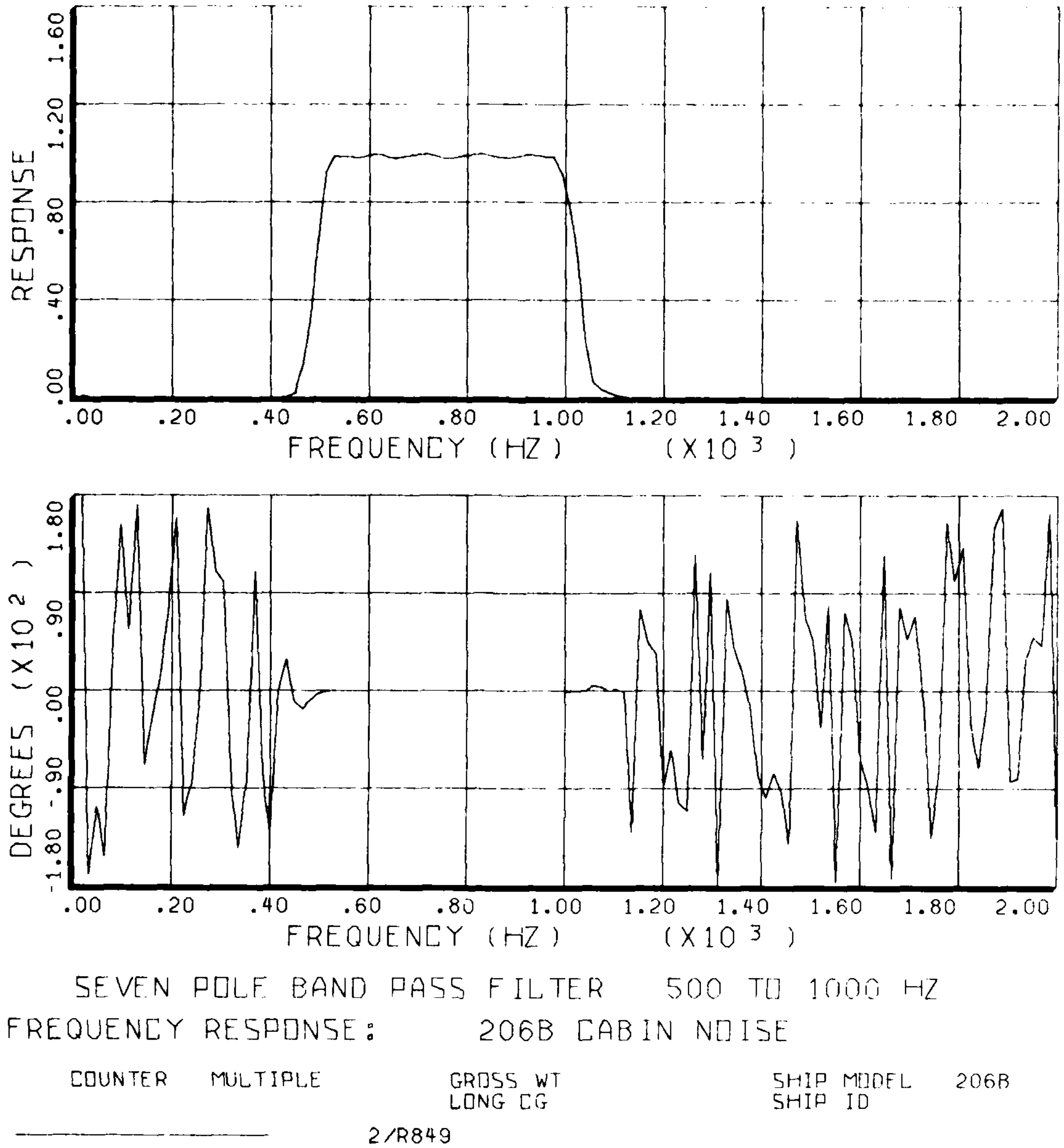
AUTO-SPECT DENSITY: 206B CABIN NOISE

COUNTER 2 GROSS WT SHIP MODEL 206B
LONG CG SHIP ID

2/R849

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/18/80

Figure 20. Auto-spectral density of acoustic data (log-log scale).



BHT.USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/18/80

Figure 21. Frequency response function for a digital filter.

Auto-Correlation of Cycle-Averaged data to take advantage of the implied periodic nature of these data. For non-Cycle Averaged input data, a special rectification method is used to compensate for the finite length of the input record. A normalization option is available to divide all calculated auto-correlation values by the zero offset auto-correlation value.

Cross-Correlation is actually calculated and labeled on output as cross-covariance, which is the cross-correlation of two time histories with their respective mean levels reset to zero. Cross-Correlation is the integral of the product of a function with another function offset by a specific time interval. Thus, the Cross-Correlation is a function of this offset. Cross-Correlation is displayed for positive and negative offsets. No window options or adjacent point averaging are provided for Cross-Correlation but ensemble averaging can be used. Both the Cycle-Averaged data algorithm and the rectification option that are available for Auto-Correlation are also available for Cross-Correlation. Normalization is also available for Cross-Correlation. Consult Section 6.1 for an explanation of normalization in this instance.

Mean is a calculation of the arithmetic mean of a time-history. Multiple time histories may be used in the estimate of mean.

Variance is an estimate of the variance of a process from one or more sample time histories.

Standard-Deviation is an estimate of the standard deviation of a process from one or more sample time histories.

Chi-Square Test for Normal Distribution uses the chi-square process to compare the distribution of sample values in one or more time histories to a normal distribution. Consult Section 6.1 for specific information or evaluation of the output from this process.

Narrow-Band Analysis is a process to be performed on acoustic data that is measured in units of ratio of pressure to the standard threshold pressure of 2×10^{-5} Newtons per square meter. This analysis uses the FFT to emulate an analog process that repeatedly applies a constant-width, narrow-band filter to an acoustic data record for a succession of equally spaced center frequency positions. The mean-square of the resultant record is calculated for each center-frequency

position. The resultant function of frequency is converted to units of sound pressure level in decibels as shown in Figure 22. A correction level in decibels may be specified by the user together with the band-width of the filter.

Octave Analysis is also a process to be performed on acoustic data and also uses an FFT approach to emulate an analog process. In this process, a bank of band-pass filters, usually eleven filters, are simultaneously applied to an acoustic record and the mean square calculated for the output of each filter. The filters are spaced in frequency using the rule that the upper break-frequency for a filter is twice the lower break-frequency and the same as the lower break-frequency for the adjacent filter that passes higher frequencies. Thus, each filter band-width is twice as wide as the next lower filter and half as wide as the next higher filter. Output is a function of octave number and is converted to sound pressure level in decibels. A correction level in decibels may be specified by the user. Consult Section 6.1 for a listing of the actual spacings of the octaves by number.

Third-Octave Analysis is similar to Octave Analysis except that each filter break-frequency is different from the next lower break-frequency by a factor of the cube root of two instead of two. Up to 32 separate third-octave levels may be calculated. Consult Section 6.1 for a listing of the actual spacing of the third-octave levels by number.

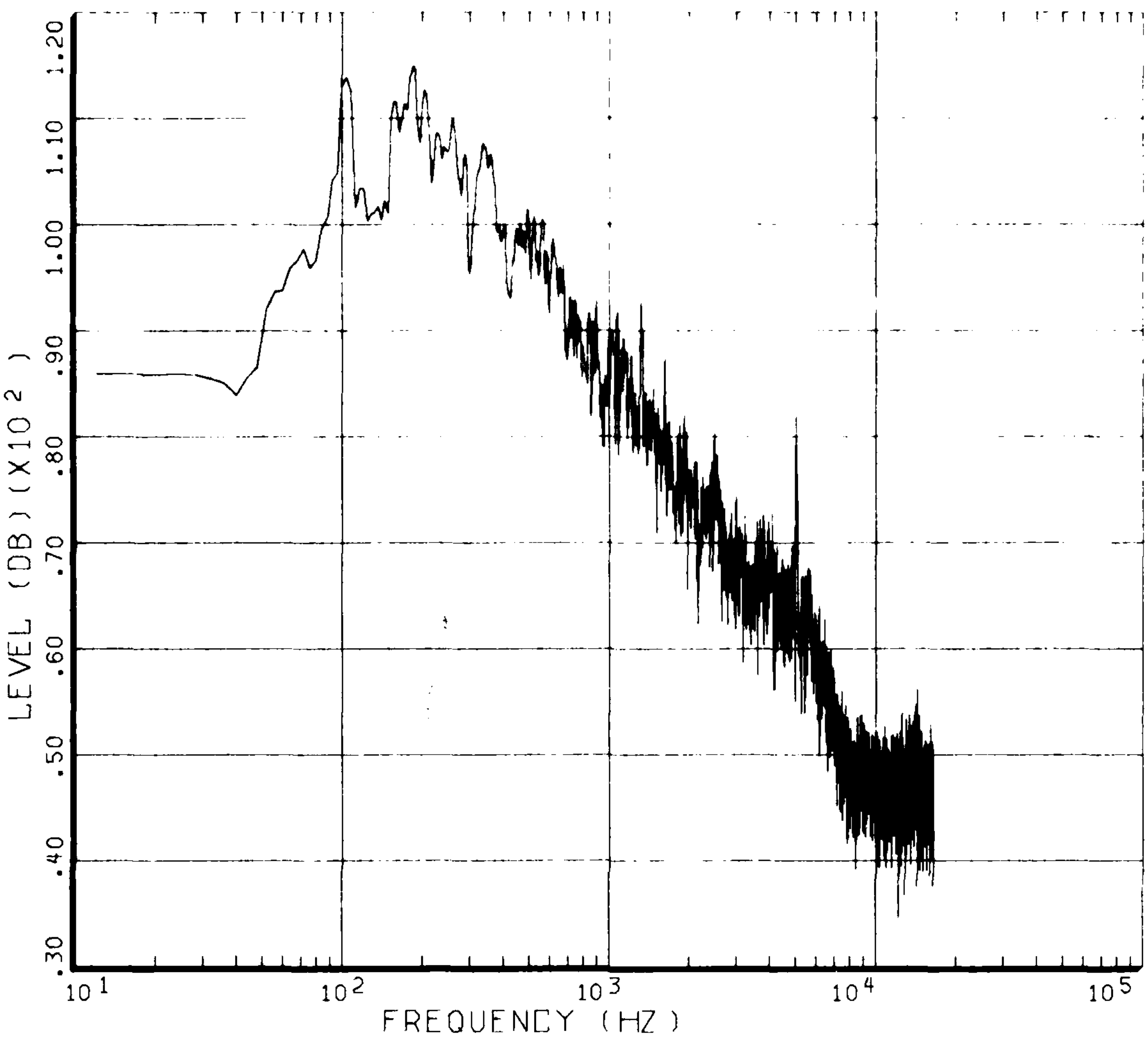
Network Weighted Integrations are performed on acoustic data to calculate the total noise level of a record with some frequency components emphasized and others deemphasized by weighing. The "A", "B", "C", and "D" weighting networks may be selected. The output from this analysis is a single sound pressure level in units of dBA, dBB, dBC, or dBD respectively. A correction level may be specified by the user.

Perceived Noise Level is a complex calculation of judged noisiness of an acoustic signal using third-octave levels. Consult Section 6.1 for a more complete description of this analysis. Output is a single level in units of PNdB for an input record. A correction level may be specified by the user.

2.5 DERIVATIONS

Following are brief descriptions of each of the derived parameter options in DATAMAP. Section 6.2 describes the method of computation for each parameter.

True Airspeed is derived from the Boom System Airspeed by smoothing, applying calibration constants, and correcting for



8 HZ NARR BAND ANAL   206B CABIN NOISE

COUNTER 2   GROSS WT   SHIP MODEL 206B
LONG CG   SHIP ID

2/R849

BHT.USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/19/80

Figure 22. Narrow band analysis of acoustic data (semi-log scale).

outside air temperature (OAT) and static pressure. Boom System Airspeed is measured in knots squared and hence requires a square root operation. For DTF input data, True Airspeed may be read directly.

Rotor Azimuth is estimated from values of time that mark the beginning and end of complete rotor cycles. These values of time are determined from the rotor azimuth pulse train and are corrected as required to account for offsets from zero degree azimuth.

Rotor RPM is the rotor speed in revolutions per minute (rpm) and is derived from the derived rotor azimuth.

Shaft Horsepower is the horsepower applied to the rotor as calculated from mast torque and rotor rpm.

Blade Static Pressure Coefficient (C_p) is the ratio of differential pressure at a blade station to the differential pressure expected on a flat blade with zero angle of attack produced by the motion of air parallel to the chordwise direction.

The Blade Normal Force Coefficient (C_n) is the integral around the airfoil of C_p components normal to the chordline. Radial position is held constant.

The Blade Chordwise Force Coefficient (C_c) is the integral around the airfoil of C_p components in the chordwise direction. Radial position is held constant.

The Blade Pitching Moment Coefficient (C_m) is the moment integral, with respect to quarter chord, of the C_p components in the chordwise and normal directions taken around the airfoil. Radial position is held constant.

Blade Displacement is the deformation in inches of the blade in the chordwise or beamwise direction for one user-specified harmonic (e.g., 2/rev). Blade displacement is calculated by integration of the corresponding harmonic component of accelerometer measurements taken in the indicated directions. Prospective users of this derivation should be certain to consult Section 6.2 regarding deficiencies in this method of calculation of blade displacement.

Blade Slope is the derivative of the Blade Displacement with respect to radius and is derived from the representation of instantaneous Blade Displacement at several radial stations.

Blade Local Flow Magnitude is the magnitude of the velocity, in feet/second, of the air flowing over a Boundary Layer Button sensor pair. Blade local flow magnitude is calculated from the two perpendicular pressure measurements taken from a single Boundary Layer Button, Boom System static pressure, and OAT.

Blade Local Flow Direction is the direction in degrees of the airflow over a Boundary Layer Button sensor relative to the chordwise direction. Positive angles indicate air moving from outboard to inboard. Blade Local Flow Direction is calculated from the two perpendicular pressure measurements taken from a single Boundary Layer Button.

Shaft Thrust Coefficient is calculated from boom system static pressure, OAT, rotor speed, and vehicle weight, or in the case of the tail rotor, antitorque force.

Shaft Torque Coefficient is derived from mast torque, boom system static pressure, OAT, and rotor speed.

Density Altitude is the altitude above sea level corrected for density variation with altitude. Density altitude is computed from OAT and Boom System static pressure.

2.6 DIMENSIONAL CAPABILITIES

Many input or output data streams specified by the user will be simple functions of time, azimuth, frequency or harmonic number. However, the Processing Program is also designed to accommodate data that may be expressed as a function of several independent variables. These variables are referred to as dimensions. The first dimension is always time or a time-related variable since DATAMAP is designed to process time histories. Frequency and harmonic number are time-related variables. When an azimuth record is available, azimuth can be associated with time for purposes of input definition, processing, or display.

Two other dimensions can be handled by the program. The second dimension and frequently the third dimension are defined from the positions of multiple sensors that measure like parameters. For example, the position of several microphones on the ground that formed a line perpendicular to the direction of travel of a helicopter could be either the second or third dimension. The independent variable could be distance

from the microphones to the intersection of the line and the helicopter path. The specification of dimensional organization of sensors that measure like parameters is performed by the Info File (see Section 4.9).

The second dimension is frequently referred to as the row position dimension. For the OLS blade application, the second (or row) dimension is the chord position. Positions in the chordwise direction are expressed as fractions of the chord length (e.g., .70 X/chord).

The third dimension is also called the column position dimension. This dimension can have either a geometric or multiple record segment interpretation. For the OLS blade application with the geometric interpretation, the third or column dimension is associated with radial position on the blade. Positions in the radial direction are expressed as fractions of the blade radius (e.g., .95 R/RADIUS).

As indicated, the third dimension could also refer to multiple record segments where each column element corresponds to a different segment of one counter or to a segment from a different counter. Multiple record segment processing is used for ensemble averaging during certain processes and to generate a significant flight parameter (e.g., True Airspeed) variation when the parameter is constant during any one counter or slowly varying during a counter that represents a long time period. For example, the program could process data from all the absolute pressure sensors at one radial station for several different counters. The output could then be displayed as a function of chord position and airspeed. Scratch files (Section 2.7) must be used to realize this capability.

The input and output data specified need not have 'extent' in all of the allowed dimensions. Frequently, multiple dimension specifications can be ignored for single functions of time, azimuth, or a time-related variable. Occasionally, as with blade local flow or blade static pressure coefficient derivations, row and column element specifications are required even though a single blade station is to be processed. Another example of zero extent in a dimension is the selection of a single time instant or azimuth position for display or processing of data from multiple row and column positions.

Frequently, the program must process two time histories for each row/column intersection. For example, considering the group of absolute pressure sensors on the OLS blade, for every chord/radius intersection there are two sensors; one sensor on the top surface and one sensor on the bottom surface. Time histories from each of these two sensors are referred to by the program as double-row elements. Data from the top surface

sensor are referred to as the 'top' double-row element and data from the bottom surface sensor are referred to as the 'bottom' double-row element. The use of double-row elements is not restricted to the above example. For instance, OLS boundary layer button (BLB) data from the inboard pointing BLB sensor are processed as the 'top' double-row element and data from the outboard pointing BLB sensor are processed as the 'bottom' double-row element.

Either one or both double-row elements may be present in processing. The number of elements on input need not correspond to the number on output. For example, Harmonic Analysis takes one double-row element time history and creates the two double-row elements amplitude and phase for output. When no double-row context is present, the 'top' double-row element is presumed in processing and the user need not consider double-rows.

The information required to specify the sensors and geometric positions of the sensors for these multiple dimensions need not be entered by the user while executing the Processing Program. These data are contained in the information file or Info File. The user need only specify the group (a subset of the Info File providing data on a particular kind of sensor) in the Info File along with the dimensional elements desired. For example, in the OLS context, suppose a user wishes to perform a process with all the absolute pressure sensors at .40 R/RADIUS (40 percent radius) radial position on the blade. The user should specify the Info File group that contains the blade absolute pressure sensors. In addition to the group specification, the user should also specify element one of the third dimension (radius variable), all elements of the second dimension (chord variable), and both double-row elements (top and bottom surfaces). All of the appropriate sensors, along with their respective geometric positions, are identified in the Info File. Refer to Section 5.8 for information on the structure and syntax of the Info File.

2.7 SCRATCH FILES

Rather than printing or plotting the results of any process, the user may choose to have the Processing Program store the output on a scratch file. A scratch file is a disc storage area that is available to the Processing Program to store and retrieve processed data. The scratch files may be a temporary disc area that is not maintained after a run of the Processing Program, or these files may be permanent so that calculations made during one run of the Processing Program are available during a subsequent run of this program. Three scratch files are visible to and accessible by the user and are referenced as SCF1, SCF2, and SCF3. Any of the three files will hold data organized in all the dimensions of processing listed in Section 2.6 within the limits of available storage.

Once a process result has been stored on a scratch file, that processed data can be used as input for a new process or simply used for display. All or part of the scratch file contents can be accessed at the user's option.

One of the primary advantages provided by scratch files is the ability to perform sequences of processes on the same data. An example is outlined in Figure 23. This example is the same as the example in Section 1.3 except that here the actual processing sequence will be given. Blade absolute pressure data for all the sensors on the OLS blade are loaded on the Master File together with airspeed, azimuth, static pressure, and outside air temperature (OAT) data. C81 simulated normal force coefficient, C_n, data for the same flight conditions are loaded on the Master File as well. In the Processing Program, cycle-averaging is performed on some integer number of rotor cycles of the blade absolute pressure data and the resultant single cycle for each of the sensors is stored on scratch file 1 (SCF1). The pressure data from the 86 percent radial station, top surface, are displayed in Figure 24. Compare these data with Figure 1. The data in Figure 24 show large high-frequency components, which shall be assumed of no interest for the purposes of this example. Thus, the data in scratch file one are digitally filtered and the result stored in scratch file two (SCF2). The low-pass digital filter selected has a break-frequency of 200 Hz and five poles. Figure 1 shows the pressure data from the 86 percent radial station, top surface, after digital filtering. Figures 2 through 5 show data from SCF2 as well.

After viewing the data in SCF2 in various ways, the blade static pressure coefficient, C_p, is calculated from the pressure data and the result stored in scratch file three (SCF3). The attached parameters, true airspeed, azimuth, rotor speed, static pressure, and outside air temperature are also required for the C_p derivation (see Section 2.8). Figures 6 and 7 show data from SCF3. The normal force coefficient, C_n, is integrated from the C_p data in SCF3 and the result stored in SCF2. The pressure data previously stored in SCF2 are destroyed. Figures 8 and 9 show plots of C_n from SCF2.

Now C81-generated C_n data are cycle-averaged and stored on SCF1. The pressure data previously stored in SCF1 are destroyed. Figures 10 and 11 show plots of C81 generated C_n from SCF1. Figure 12 shows a direct comparison of measured

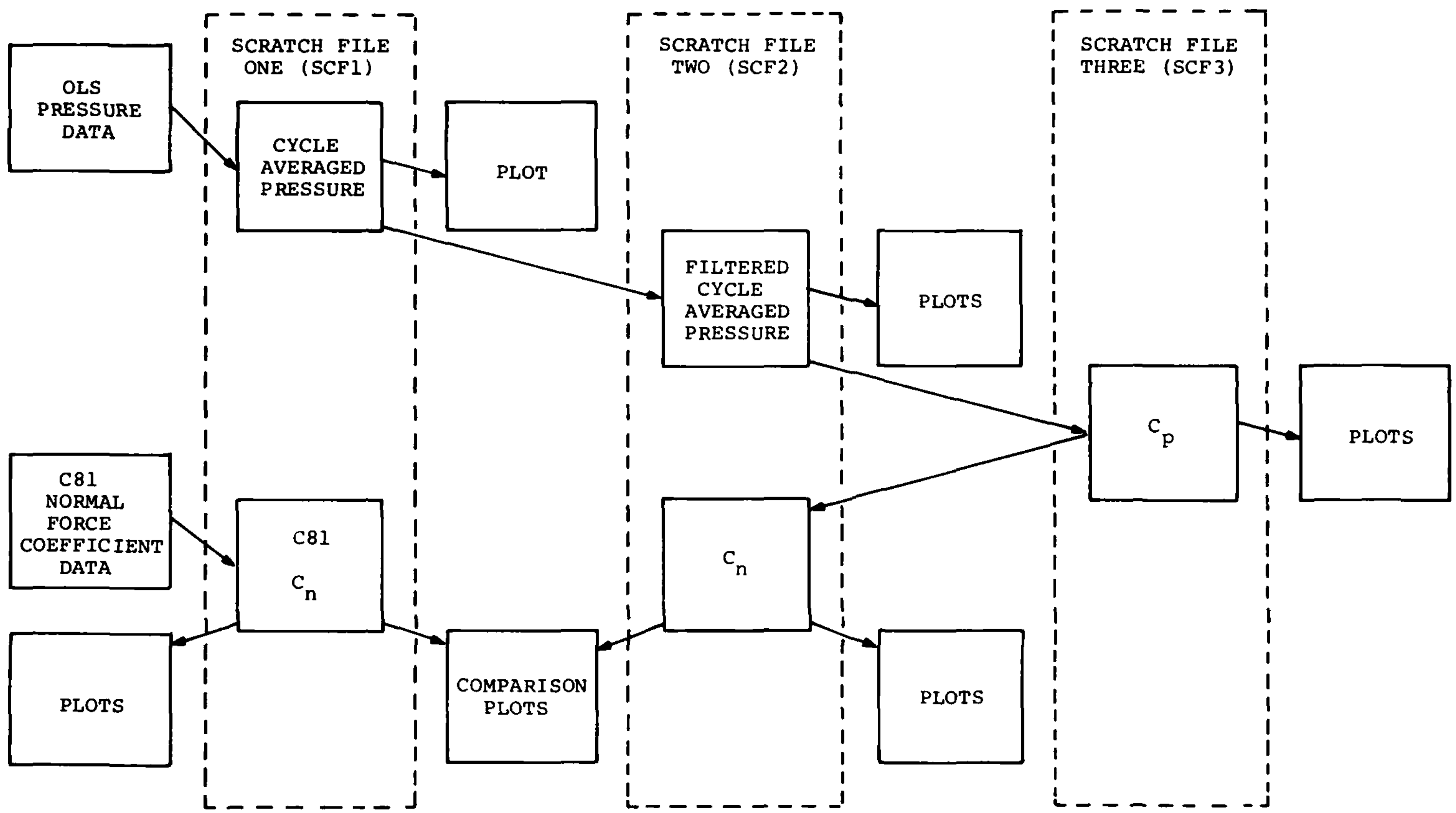


Figure 23. Linking processes with scratch files.

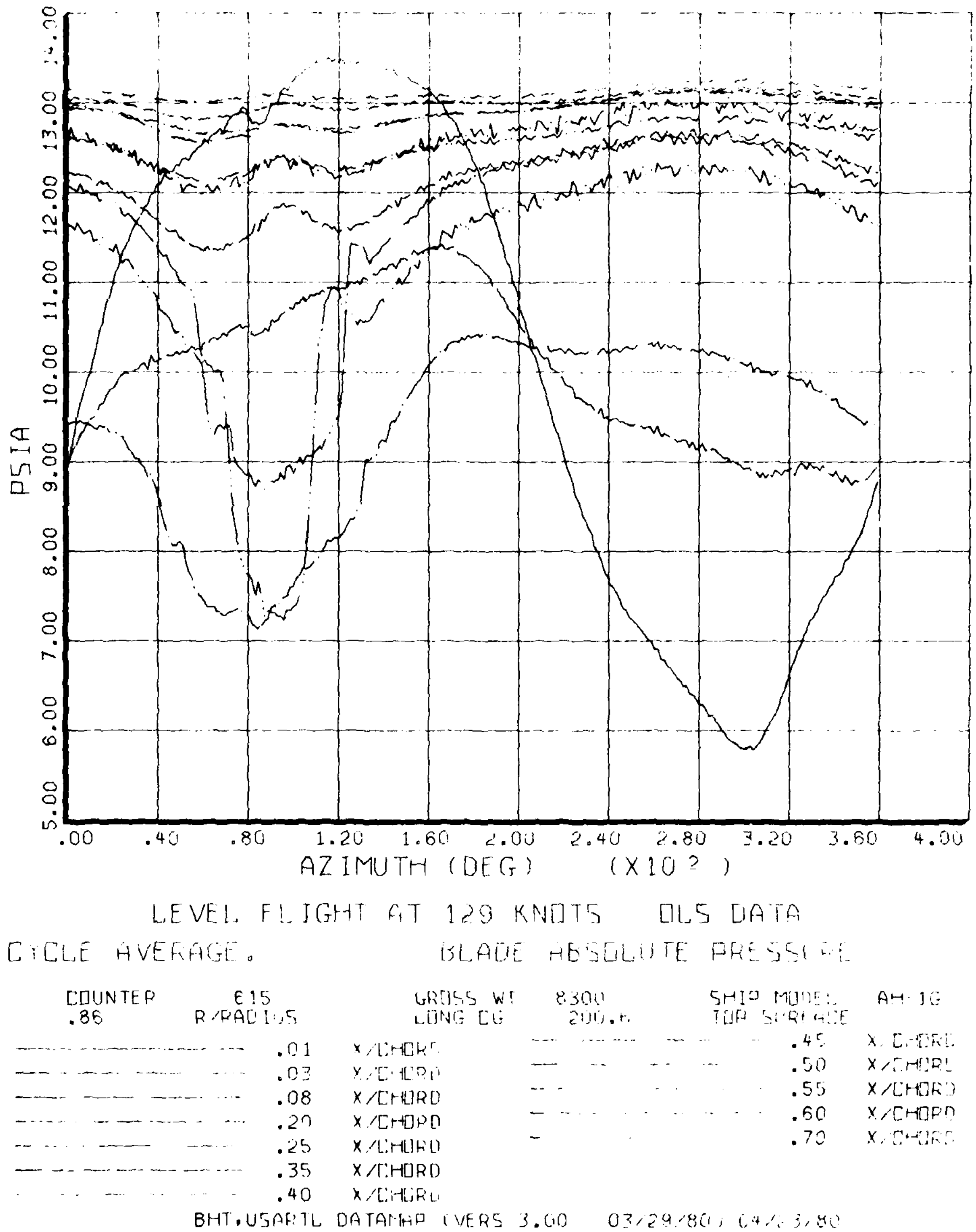


Figure 24. Cycle-averaged blade absolute pressure for one radial station.

C_n and C81-generated C_n for radial positions that are nearly the same.

Note that in the above example the intermediate results of a sequence of processes are accessible to the user. If at any time a process yields an unusual result, the user may interrogate the input to the process, which was the output of a previous process. For example, if a user discovers questionable values of C_n, the input C_p data may be displayed to gain additional insight.

In the above example, current values stored in a scratch file were overwritten and destroyed when new data were stored on the file. It is possible to add data to a scratch file without destroying the data already present. In order to add data to existing data on a scratch file, the data being added must be of the same parameter and sample rate as the data already present. Otherwise, results are unpredictable. Also, the data being added can represent only one column (radius) position although it can have multiple row (chord) positions and double-row elements. This restriction is present because each ADD to a scratch file is considered to be a third-dimension element or column.

The capability to ADD data to a scratch file is useful when generating plots that contain data of more than a single counter. If the user wished to compute Shaft Torque Coefficient as a function of airspeed, data from several counters with differing airspeeds but otherwise like conditions could be used. Shaft Torque Coefficient may be computed for each counter separately and the results added to SCF1. Plots of Shaft Torque Coefficient versus airspeed may then be generated.

2.8 ATTACHED PARAMETERS

Certain variables are required very frequently for analyses, derivations, and displays. These variables are called attached parameters. Attached parameters include rotor azimuth, true airspeed, rotor rpm, Outside Air Temperature, and Boom System static pressure. Attached parameters are assumed to be slowly varying and are smoothed in derivation and stored at the rate of one sample per rotor cycle.

Attached parameters are calculated whenever required for a process, a plot scale, or when the output is to a scratch file. The values are maintained in the memory of the computer so that they need not be recalculated when a new step references the same counter and time period. When results are written to a scratch file, all of the attached parameters are

also stored. These parameters are then available for subsequent analyses, derivations, or plots involving the stored results.

Manipulation of the attached parameters is transparent to the user with three exceptions. First, the user must assure that the proper item codes are present on the Master File partition to provide for the derivation of rotor azimuth, indicated airspeed, outside air temperature and Boom System static pressure. Second, the user has the ability to specify, for some derivations, a constant value for OAT and/or static pressure. Finally, the user may need to enter airspeed calibration constants for the true airspeed derivation (see Paragraphs 5.3.2 and 6.2.2).

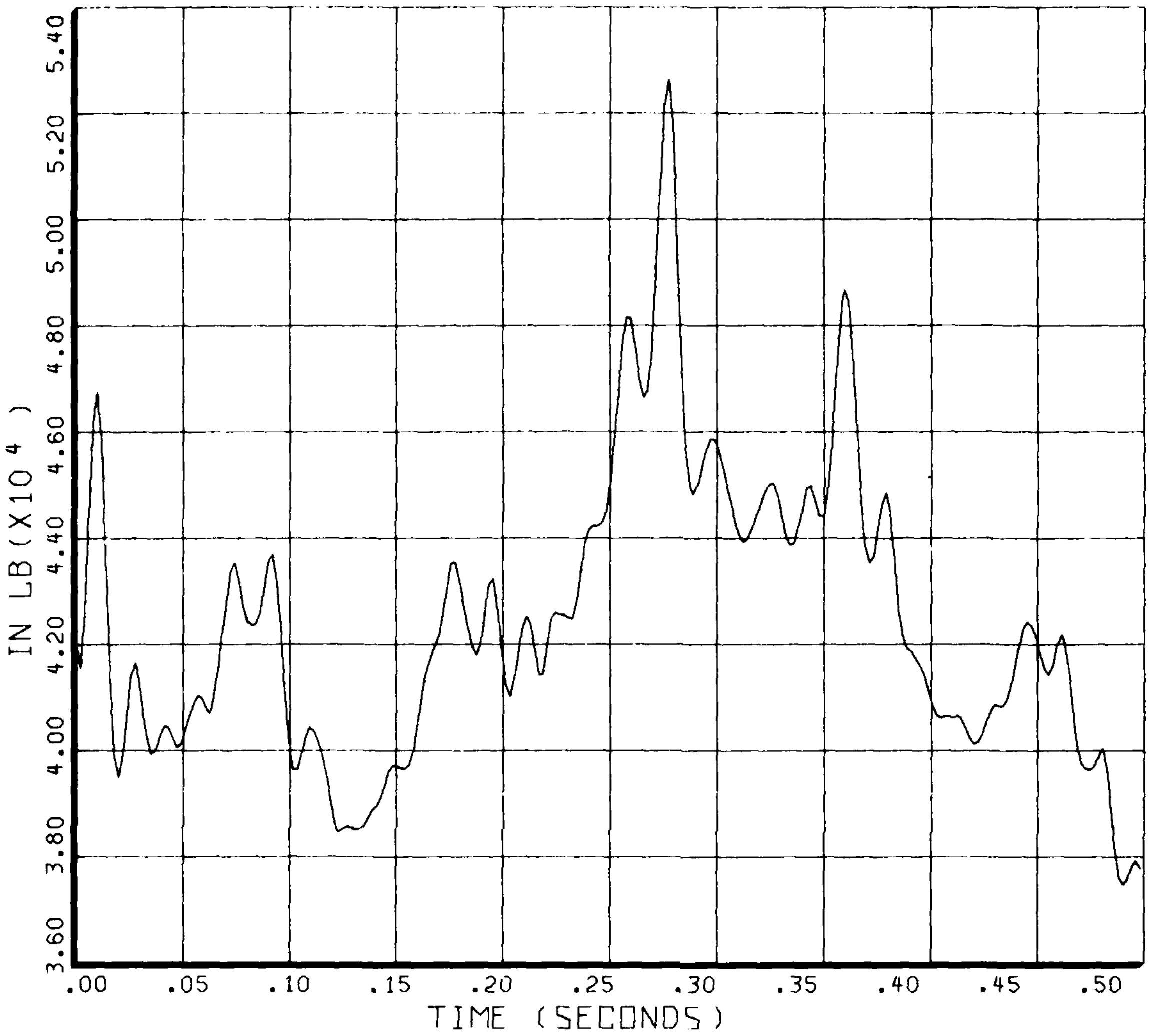
2.9 <u>OUTPUT CAPABILITIES</u>

Processed results or raw data can be displayed by the Processing Program in two basic forms: printout and plots. Printout consists of tabular listings of one or two double-row elements along with time, or a time-related variable. For outputs with more than one second- or third-dimension element, each row/column intersection forms a new tabular listing. Listings generated in the batch mode of operation are routed to a line printer, while listings generated in the interactive or interactive graphics mode are routed to the interactive terminal.

Three general kinds of plots are available: X-Y plots, surface plots, and contour plots. X-Y plots depict functions of one variable in Cartesian coordinates with dependent variable and independent variable axes. The increment between annotated points on each axis is restricted to powers of ten multiplied by the integers 1, 2, 4, or 5. Figure 25 shows an example of a simple X-Y plot. Four specific types of X-Y plot may be selected: simple X-Y plots, multiple-curve X-Y plots, comparison plots, and double-scale X-Y plots.

Multiple-curve X-Y plots are essentially the same as simple X-Y plots except that more than one function may be shown on the same plot. Curves on the plot are differentiated by various sequences of dashed or dashed-dotted lines. Samples of the dashed lines are labeled below the plot. However, these labels are of very limited length so that differences between the curves should be describable as different values of a single parameter such as row element position (as shown in Figure 1), column element position, azimuth, or counter. The several curves on a multiple-curve plot may all be drawn at one time or they may be added one at a time. New variables must agree in kind with the variables shown on the original plot scale.

Comparison plots also allow more than one curve on an X-Y



800 FPM DESCENT AT 37 KNOTS

TIME HISTORY: MR MAST TORQUE

COUNTER	956	GROSS WT	8300.	SHIP MODEL	AH-1G
		LONG CG	196.3	SHIP ID	20391

956/M107

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/19/80

Figure 25. A simple X-Y plot.

plot. However, as shown in Figure 12, the annotation for each curve is considerably more extensive. Comparison plots are particulary useful for display of test and simulation data on the same graph. Curves must be drawn on a comparison plot one at a time. Depending upon the amount of commentary and annotation, there is annotation space for no more than three or four curves on each comparison plot.

Double-scale X-Y plots are the same as multiple-curve X-Y plots except that the plotting area is divided into two plotting areas with separate dependent-variable scales. Figures 7 and 21 are double-scale X-Y plots. The two plotting areas are for the top and bottom double-row elements as described in Section 2.6.

The user has the option to specify a logarithmic scale for the independent and/or dependent variable of an X-Y plot. With a log scale specification, the user must indicate the number of decades to depict for the axis in question. The highest decade on an axis will always be the one that includes the largest value present for the corresponding variable. Zero and negative values that occur in a variable to be plotted with a log scale are reset to a very small positive number. Figures 19, 20, and 22 provide examples of log scales.

Linear independent and dependent variable scales are normally scaled automatically to include all values on the plot. However, the user has the option to specify either scale size by specifying the minimum value present and the increment between annotated points. Nine increments, or divisions, are provided on the vertical axis, while ten increments are provided on the horizontal axis. The increment must be one of the integers 1, 2, 4 or 5 multiplied by a power of ten. Similarly, the minimum value must be a multiple of the increment. Values in the function that exceed the user-specified scale will be clipped. Figure 26 shows a plot with a user-defined scale.

For double-scale X-Y plots, the dependent variable axis may be specified separately for each plot. In addition, a logarithmic scale may be specified for one dependent variable axis and a linear scale specified for the other dependent variable axis.

When X-Y plots are to be drawn on the Tektronix 4014, the user may specify that the crosshair graphics cursor be activated immediately after the plot is completed. The crosshairs may be used to evaluate points on the screen in user coordinates.

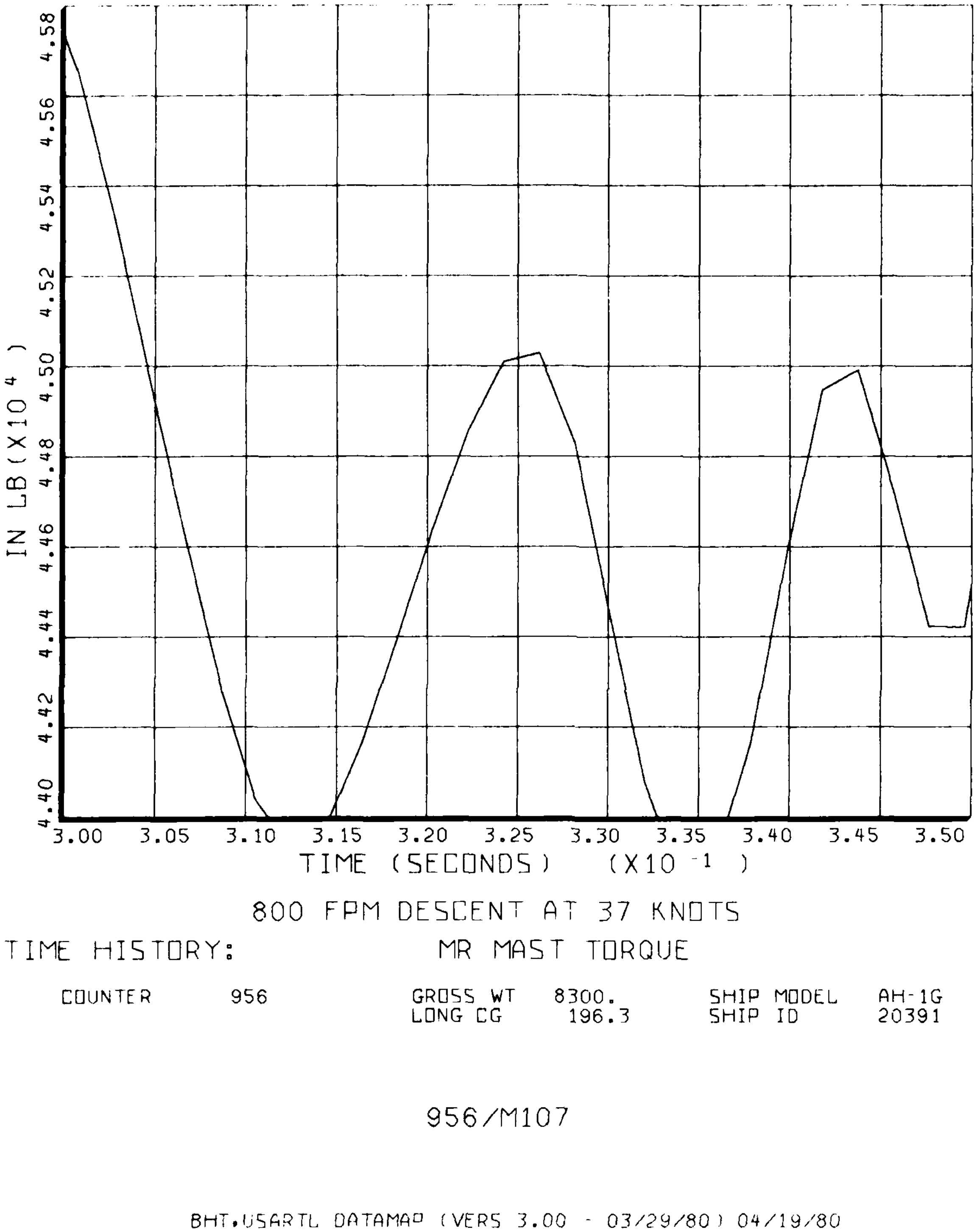


Figure 26. An X-Y plot with user-defined scale.

Surface plots pictorially depict a function of two independent variables. The user is provided a perspective view of the surface that is represented by a set of curves along each independent axis. Each set of lines is equally spaced and orthogonal to the other independent axis. The coordinate system for surface plots can be rectangular or cylindrical. For cylindrical plots, one independent variable must be a complete cycle of rotor azimuth. Figure 4 shows an example of a surface plot in cylindrical coordinates, while Figure 3 shows a surface plot in rectangular coordinates.

Algorithms used to formulate surface plots permit the user to observe the surface from essentially any point in three-dimensional space. The position of the observer in 'X', 'Y', and 'Z' is specified in units of the maximum width of the surface. The observation point must not be located inside the object. Observation points located within one object width from the center cause undesirable distortions in the plot.

Contour plots depict a function of two independent variables by drawing lines of constant dependent variable values. These lines are differentiated for the user by the following scheme. Contour levels are equally spaced and the contour lines for every fourth level are solid with letters superimposed on the curves. Samples of these lines are drawn below the plot and labeled with the function value. Three intermediate contour levels are represented by dashed lines. The length of the dashes decreases with increasing contour level. Contour plots can be drawn in rectangular or cylindrical coordinate systems. For cylindrical plots, one independent variable must be a complete cycle of rotor azimuth. Figure 2 shows a contour plot in rectangular coordinates, while Figure 5 shows a contour plot in cylindrical coordinates.

For contour plots, the user has the option of specifying the minimum contour level of interest, the increment between levels, and/or the maximum number of contour levels that can be drawn. Normally, the minimum contour level and the increment between levels are selected by an algorithm similar to the method used for selecting scale values on X-Y plots. Contour level values are always 1, 2, 4 or 5 multipled by a power of 10. The user may specify a minimum value and increment that is consistent with these rules but that may increase (or decrease) the number of levels for some range of contour values.

When the maximum radius value provided for a cylindrical format contour plot is between .85 and 1.00 (regardless of units), the plot is drawn with the outside circle at 1.00 and

the space between the maximum radius and the outside circle is left blank (see Figure 5). This feature is included so that the blade radius will be apparent when the maximum radius value is inside the blade tip and to facilitate comparison of simulation and test data (compare Figures 8 and 10).

For incremental plotter output, the graphs are designed to fit on preperforated 8-1/2 by 11-inch paper. Each plot frame is translated 8-1/2 inches to the right of the previous one. This 8-1/2-inch-frame increment may be modified by the user at the beginning of a run of the Processing Program.

For Tektronix output, the plots are drawn on the right side of the screen and occupy approximately 60 percent of the screen width. Prompting messages from the program and inputs from the user are written on the left side of the screen (Figure 27). Tektronix plots are nearly identical in size and form to corresponding incremental plotter graphics. The plotting area can be expanded on the Tektronix to include all the space available on the screen, with proportion maintained, so that only the axis annotation is shown and other plot labeling is left off. No corresponding plot expansion capability exists in batch mode.

Two graphics enhancement capabilities are available only on certain installations of DATAMAP. The first of these capabilities is the quick plot mode. In this mode, lines of character annotation that are oriented horizontally are output as alphanumeric data rather than stroke-generated characters. Plot annotation is generated considerably faster in this mode.

The second of these capabilities is the plot copy mode. In this mode, plots can be generated in the interactive or interactive graphics modes and then copied, at the users option, to a holding file. This file may then be plotted on the same graphics device that is addressed by the batch version of the Processing Program. Naturally, plots generated in the interactive mode never appear until they are actually plotted on the batch graphics device. Use of the plot copy mode can degrade plotting speed in the interactive graphics mode.

DATAMAP has the capability to convert units for the dependent variable on output. Unit conversions are specified in the Info File (see Section 5.8) so that a unit conversion is applicable throughout a run of the Processing Program. Any linear conversion may be specified.

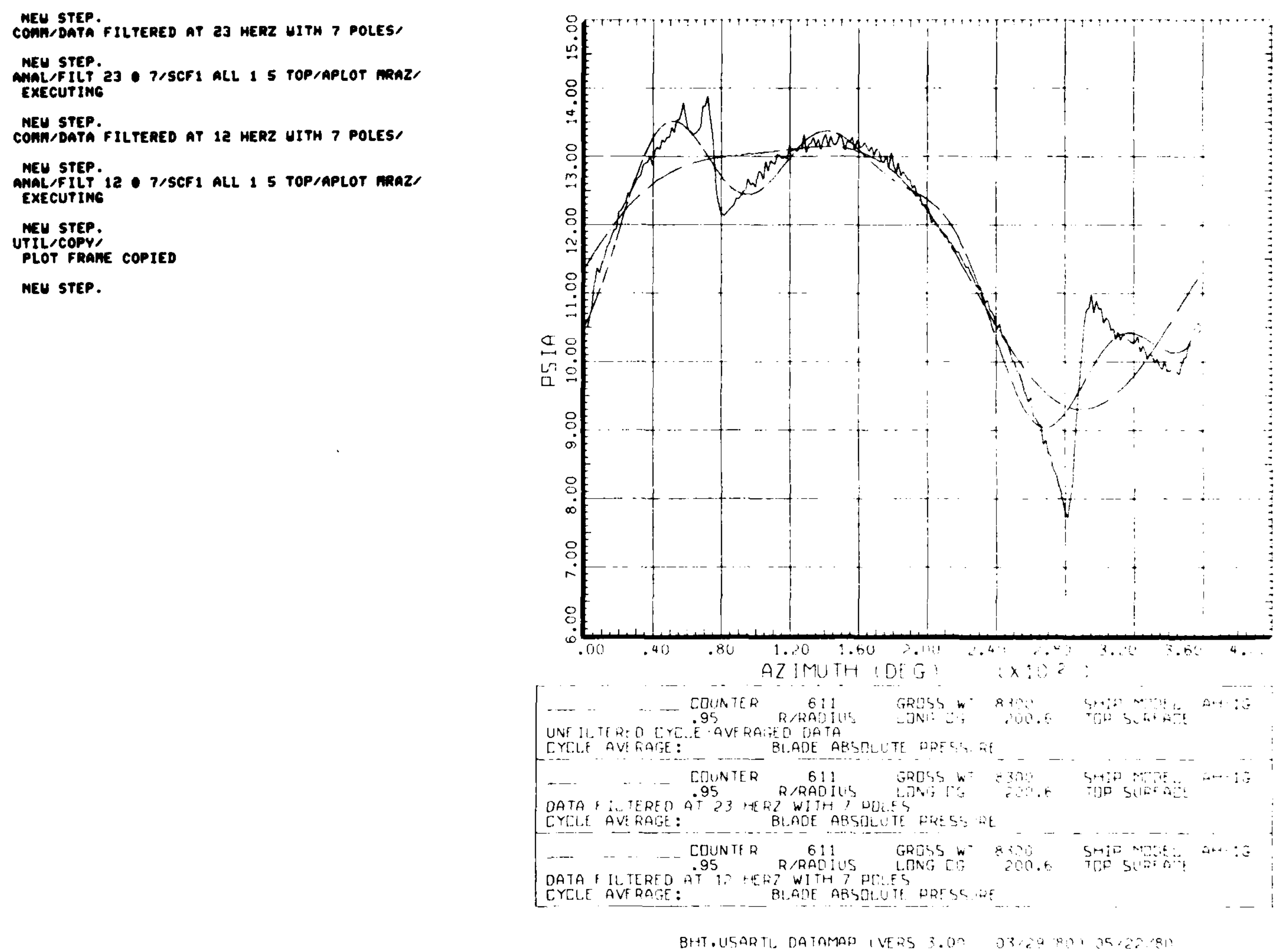


Figure 27. Tektronix screen apportionment.

3. DATA FILE CREATION/MODIFICATION

3.1 MASTER FILE INITIALIZATION

Assuming that the space for a Master File exists on a direct access device (i.e., disc), it must first be initialized before data can be stored on it. Specifically, for IBM system application, the initialization process is to write zeroed records on every record position in the direct access file. If an initialization capability was included in the File Creation Program as an option, a user might inadvertantly initialize the Master File when it contained useful data.

The Master File Initialization Program performs the function of writing on every record in the file. This program then presets the first, or directory record, for the Master File with the following information: the total number of records in the file, a password for the entire file, and a list of partitions showing that none are present. This program requires one line of user input in free-field format containing two entries that may appear in either order. One entry is an integer specifying the size of the Master File in records (currently 256 words or 1024 bytes per record). The other entry defines the password for the entire Master File called the superword. The purpose of the superword is explained in Section 3.4.

If the disc space allocated for the Master File is insufficient for the number of records specified by the user, the program will terminate abnormally when it attempts to write on a record that it cannot access. While running, the program lists every 50th record it is attempting to write on so that, in the event of an abnormal termination, the user will have an estimate of how many records were successfully written. If the intialization is successful, the program will print the message (for example):

```
FILE INITIALIZATION SUCCESSFUL
RECORDS = 20800
SUPERWORD = BIGWORD
```

The user can then proceed to submit batch runs of the File Creation Program to transfer data to the Master File.

3.2 FILE ADDITIONS

3.2.1 Characteristics of the File Creation Program

The File Creation Program (FCP) transfers data from BHT-GDC format digital tape or from a DTF to the Master File. Data selection for transfer and control parameters for this program is supplied by user input, which must be in a format explained in Paragraph 3.2.3. This input specifies item codes and counters to be transferred along with other information.

The program attempts to transfer all the item code/counter combinations for all the item codes and counters specified. If an item code/counter pair is not found in the available input data, the corresponding data are simply left off the Master File partition. Data that are found are still written on the partition even though other requested data cannot be located on the input medium. Sometimes the amount of data available for a specific item code/counter is less than the time history length requested. In this case, the truncated data record is transferred.

When counters are specified, a time history length and offset from the start of input data to the start of the time histories to be transferred is associated with each counter. This offset and length applies to every time history (item code/counter pair) corresponding to that counter. Each item code specified has associated a sample rate reduction factor or an arbitrary sample rate for DTF data input, a digital filter breakpoint (-1.0 implies no filtering), and a format for data storage on the Master File (calibrated or uncalibrated). DTF data must be calibrated. These factors will apply to every time history corresponding to that item code.

The File Creation Program can operate in three different modes with regard to the partition that is receiving the data. In the NEW mode, a new partition with a new partition name is created. In the ADD mode, data are transferred to an existing partition without disturbing data already present except that duplicate item code/counter pair time histories are overwritten. In the REPLACE mode, the data on an existing partition are deleted and new data are written into the partition with the original partition name retained.

In the NEW mode, a password is set for the partition. The same password must be entered in any subsequent run of the File Creation Program using the ADD or REPLACE mode for the same partition. The intent of this security feature is to prevent inadvertant rather than intentional destruction of data. The Master File monitor (see Section 3.5) has the capability to override the password, using the superword capability.

After all the data specified in the user input have been transferred, or after the data input medium (digital tape or DTF) is exhausted, the File Creation Program can provide two services for the user. The first of these services is the generation of a Map, or listing, of the contents of the partition. The Map contains a list of counters present and, for each counter, a list of the item codes present. For each item code, the following information is included: record length, sample rate, digital filter applied (-1 indicates no filtering), specified offset, time alignment offset, item description, and item units. If the user wishes to suppress the Map, then the instruction 'NOMAP' must be specified.

The second post-transfer service that the File Creation Program can perform is a SAVE of the partition. When the user specifies SAVE in the input, the program will copy the partition to digital tape. The SAVE process is the same as the partition SAVE used in the File Maintenance Program (see Section 3.4). The user has the responsibility to assure that the partition will fit on one digital tape (see Section 3.4). Restoration of a partition copied in this fashion must be performed with the File Maintenance Program. The user must specify SAVE in the input when a partition copy is required since SAVE is not a default entry.

Computer CPU time limit settings are critical in the operation of the FCP. If the program runs out of time before transfer of data to a partition is complete, then that partition will certainly be lost. In some cases, the entire Master File could be destroyed if a time limit specified were too short. The user must assure that a generous time allotment is specified for each run of the File Creation Program when data already exist on the Master File.

Unfortunately, it is impossible to provide the user with a precise method for estimating the computer CPU requirement for a run. Time requirements can change considerably as computer CPU models and local installation factors are changed. Time requirements can also change as processing factors (e.g., digital filtering) are altered. For example, on the IBM 370/168 installed at BHT, about five minutes were required to create a partition of 15,000 records from 16 digital tapes using no digital filtering. Digital filtering might have increased the time requirement substantially. Sample rate reduction would reduce the filtering time and also reduce the size of the partition. If the total Master File space was 30,000 records of which 15,000 were already occupied, then an additional zero to five minutes could be required to rearrange the partitions in the Master File to create the maximum space for the new partition. The user must gain experience with the CPU time usage for this program while being extremely generous in time limit specifications.

3.2.2 Setup Actions Required

Before attempting to build an input data set for the File Creation Program, the user must gather certain information. The item codes and counters that specify the data necessary for processing must be determined. For BHT-GDC format input, the user must then find out which digital tapes contain these data and the sample rate for each time history to be transferred. For each counter selected, the user must determine the record length for the time histories to be transferred, and a time offset between the start of data on tape or on the DTF and the beginning of the data to be transferred. The time offset may be zero or any positive time interval that is less than the length of the corresponding time histories stored on tape or on the DTF. The record length for the time histories to be transferred is bounded by the length of the time histories on tape less any time offset applied.

For each item code, the user must calculate the sample rate modification to be applied and decide whether the data should be calibrated and/or filtered (filtered data must be calibrated). If, for example, the user wishes to perform calculations using multiple sensors of the same type (e.g., blade absolute pressure sensors to calculate C_n), then the user must assure that the sample rates for all these sensors are identical for all the corresponding time histories stored on the Master File partition.

For DTF data input, the user has the option to specify that all item codes and/or all counters will be transferred. If all counters are specified, then the same offset and record length is implied for every counter. Similarly, if all item codes are specified, then the same digital filtering and sample rate modification is implied for every item code. For DTF data input, the sample rate of data to be stored on the Master File may be specified arbitrarily. If the sample rate of the data stored on the DTF is constant, a sample rate reduction factor may be specified instead. DTF input data do not require a selection between calibrated or uncalibrated since data stored on a DTF are already calibrated.

The user should then estimate the number of records on the Master File required to store these data. Four kinds of records are required to store these data: data storage records, time history information records, partition directory records, and partition records. Eight partition records are always required. The following is the formula for determining the number of records required for storage of data on the Master File.

$$A = N \times M$$

$$B = \sum_{i=1}^{N} \sum_{j=1}^{M} [R_i L_j / K]$$

$$C = [M/128] + M \times [N/128]$$

$$S = 8 + A + B + C$$

where

N = number of item codes

M = number of counters

R_i = final sample rate for item code "i".

L_j = final record length for counter "j".

K = 256 if data are calibrated or 512 if data are uncalibrated.

S = number of Master File records required for storage of the data.

This formula is valid only for the initial storage of data on a partition, or for a complete requirement of a partition.

This estimation procedure is provided for use when Master File space is restricted. If space is plentiful, then a rough estimate of space required should be adequate. Rough estimates should be generous since the File Creation Program will not exceed the specified partition size (SPACE entry). When all the requested space is not used, the partition produced will include only the records necessary to store the data. Any surplus storage area is returned to the available pool.

3.2.3 User Instructions

Input to the File Creation Program will consist of a sequence of card images or lines, each line containing one or more entries. An entry is a sequence of alphanumeric characters unbroken by commas or blanks. The slash (/) and single quote (') characters must not be used in an entry. A comma or any number of blanks separates the entries. Input is regarded as a continuous sequence of entries from the first entry on the first line to the last entry on the last line. However, a single entry cannot span two lines. The entries on each line must be restricted to the first 60 columns.

There are two broad categories of entries, literal and numeric. The first character of a numeric entry is '+', '-', '0-9' or '.'. A numeric entry is free field and is interpreted as integer or floating according to context. Thus, decimal points are not required for floating input unless necessary to include a fractional part. Exponential entries (e.g., .1260E-5) are not allowed. Literal entries begin with any characters other than those that begin numbers and are interpreted as a character string (e.g., FILTER, JOHN-SMITH, PRESSDAT).

The first kind of literal is an 'instruction word.' An alphabetical list of instruction words appears with individual explanation later. Instruction words can specify instructions by themselves or may require a qualifying string or numeric entry to complete the instruction. Instruction words have preassigned meanings that control the input.

A second kind of literal is the qualifying string mentioned above, e.g.,

```
NEW PRESSDAT
```

'PRESSDAT' would be a partition name and is the qualifying string.

The final kind of literal is an item code. Any literal that is not an instruction word or qualifying string and that is exactly four characters long is assumed to be an item code.

The first kind of numeric input is a qualifying number, a value that must go with an instruction word. For example,

```
SKIP 32
```

means record only every 32nd data point read from tape. A number that is not a qualifying number is assumed to be a counter if it is in the range 1-999999.

A description of each instruction string follows. Note that initial letters of each word are underlined. These are the minimum number of letters required to define the word. Additional letters up to a total of four must match, others are ignored. Thus, CA means the same as CALIBRATE, which could also be written: CALIXXX, CALI or CAL but not CALX.

ABSOLUTE (number) — Absolute offset. 'Number' seconds of data (floating) will be discarded starting with the first point found regardless of time skew alignment requirements. This value is associated with all counters input

after the instruction until a new ABSOLUTE or OFFSET command is found.

ADD (literal)	Add data to a partition. The data to be input will be added to a partition of data already present with name identical to the literal.
ALIGN	System will attempt to align first points recorded so that every item code begins at the same time. This instruction may appear anywhere before the END instruction except where it would affect the context of other instructions.
ALL	This keyword can be used only for DTF input. The next entry must be ITEMS or COUNTERS. If ITEMS is the next entry, the program will attempt to transfer data for every item code found that matches the specified counters. If COUNTERS is the next entry, the program will attempt to transfer data for every counter found that matches the specified item codes. Both ALL COUNTERS and ALL ITEMS may be specified in one run. ALL COUNTERS should not be used in conjunction with any individual counter listing and ALL ITEMS should not be used in conjunction with any individual item code listing.
CALIBRATE	Store calibrated data. Data for all item codes that appear after this instruction and before any subsequent NOCALIBRATE command will be stored calibrated. This instruction does not affect DTF data, which are calibrated as stored on the DTF.
COUNTERS	If this keyword immediately follows the ALL keyword for DTF data input, the program will attempt to transfer data for every counter found that match the specified item codes. Otherwise, this keyword is ignored by the software and may be used as a comment to indicate that counter information follows.
END	End of input. Any entries that come after the END will be ignored.

FILTER (number)	Low-pass filter breakpoint. Indicates the upper breakpoint in hertz for a filter to be applied to every item code that appears after this instruction and before a subsequent FILTER or NOFILTER instruction. Filtered data will be calibrated regardless of CAL or NOCAL instructions.
ITEMS	If this keyword immediately follows the ALL keyword for DTF data input, the program will attempt to transfer data for every item code found that match the specified counters. Otherwise, this keyword is ignored by the software and may be used as a comment to indicate that item code information follows.
NEW (literal)	New partition. The data to be input will be a new partition with name given by the literal. The partition name may include from 1 to 8 characters.
NOFILTER	No filter will be applied to data from item codes that appear after this instruction and before a subsequent FILTER instruction.
NOCALIBRATE	No calibration will be applied to data from item codes that appear after this instruction and before a subsequent CALIBRATE instruction. This instruction does not affect DTF data, which are calibrated as stored on the DTF.
NOMAP	After partition generation or modification is complete, a list of item code-counter pairs present in the partition is normally generated. NOMAP suppresses this list.
OFFSET (number)	Offset from aligned value. Offset 'number' seconds (floating) after alignment. Equivalent to ABSOLUTE if ALIGN is not specified.
PASSWORD (literal)	If NEW, enter (literal) as the password for the partition. If ADD or REPLACE, enter (literal) to allow the modification of the partition. The password may include from 1 to 16 characters.

RATE (number)	Sample rate for data to be stored on the Master File. This keyword may be specified only for DTF input. Interpolation will be performed on input data to produce the required sample rate. Applies to item codes that appear after this instruction and before a subsequent RATE or SKIP instruction.
RECORD (number)	Amount of data read. System will attempt to read 'number' seconds of data (floating) for the counters that appear after this instruction and before a subsequent RECORD instruction. The number of points transferred depends on the SKIP factor (below) and the sample rate for the data as stored on tape.
REPLACE (literal)	Replace a partition. The partition (literal) will be deleted and a new partition with the same name will be built.
SAVE	After partition generation/modification is complete, this command will generate a tape save of the partition (the tape must be specified in the JCL as FT15F001)
SKIP (number)	Skip factor. Reduce the number of points transferred by using only every 'number' point in the sequence. A skip factor of one means no skipping. Applies to item codes that appear after this instruction and before a subsequent SKIP or RATE instruction. A skip factor may be applied to any BHT-GDC format data and to DTF format data that have a constant sample rate.
SPACE (number)	Storage space allowed for partition. Number of random access records (1024 bytes/record) to be allowed as maximum size for the partition. For an ADD, this limit includes the space already used by the partition before additions.
STRANGE	Specifies that the data input tape format will be other than the standard BHT format or DTF format.
TAPES (number)	Number of data tapes to be scanned for data. This keyword should not be used for DTF input.

THRU (number) — Requires counters to be entered before and after this entry and specifies that all counters between those entered before and after the 'THRU' entry will be transferred. Thus,

28 THRU 32

means counters 28, 29, 30, 31, and 32 will be used.

USER (literal) — Enter user name. (Literal) will be substituted as the partition user name if the password is correct. The user name may include from 1 to 16 characters with no blanks.

ORGANIZED EXAMPLE -

```
NEW BLADESET USER J-SMITH PASS ABCD
ALIGN SPACE 2000 TAPES 2
ITEMS
CAL SKIP 4
R123 R124
NOCAL SKIP 8
R281 R282 R283
FILTER 16 SKIP 32
R242 R243 R244
COUNTERS
OFFSET .25 RECORD 2.0
500 512 581 616 THRU 620
OFFSET 0 RECORD .5
782 783
ABS 1.2 RECORD 1.0
10 12
END
```

DISORGANIZED EXAMPLE
(Will give same result)

```
CA SK 4 R123 R124 NOC SK 8
R281 R282 R283
FIL 16 SK 32 R242 R243 R244
NEW BLADESET PASS ABCD US J-SMITH
OFF .25 REC 2.0 500 512 581 616 THRU 620
OFF 0 REC .5 782 783 ABS 1.2 REC 1.0
10 12 SPAC 2000 TA 2 ALI END
```

Certain requirements for order in the entry sequence, or lack thereof, are evident from the entry descriptions and the examples. The END entry must appear last. Qualifying numbers and strings must immediately follow the keywords that they qualify. Counters specified for transfer must follow record length and offset instructions that apply to those counters, and they must precede record length and offset instructions that apply to other counters. Similarly, item codes specified for transfer must follow digital filtering, sample rate modification, and calibration instructions that apply to those item codes, and they must precede filtering, sample rate, and calibration instructions that apply to other item codes. Otherwise, the sequence of entries is not restricted. An organized sequence of entries is easier to read and less prone to error however.

A special line of three entries is required for DTF input. This line, called the DTF line, is considered to be separate from other FCP input and must be the first line in the FCP input sequence if it is present. The first entry for the DTF line must be the keyword "DTF." The second entry must be one of the three keywords, "LOCAL," "INCLUDED," or "SCAN." LOCAL indicates that the balance of the instruction input lines for the FCP follow the DTF line in the current system input device. INCLUDED means that the balance of these instruction lines are included in the DTF itself. Thus, the instructions that specify transfer of data from the DTF may be in the DTF as well. This option is provided to optimize the sequence of job steps that are used to run a simulation program and then execute the FCP to create a partition on the Master File. If LOCAL is specified, any instructions contained within the DTF are ignored. Instructions contained within the DTF should never include a DTF line. A "SCAN" entry means that all subsequent command input instruction lines are ignored and the DTF will be scanned for correct structure and format without transfer of data to the Master File.

The third entry on the DTF line also must be one of two keywords, "INTERNAL" or "EXTERNAL." These keywords define the format of the DTF, which is described in Volume II of this report. The format for the DTF will usually be INTERNAL. Consult local installation documentation to determine the proper specification for a particular application.

A DTF line should never be included for non-DTF (e.g., BHT-GDC format) data input. In addition, DTF input and non-DTF input should never be mixed for one run of the FCP.

3.2.4 Sample Rate Reduction and Filtering Considerations

The sample rate, or the number of equally spaced data points per second, for data stored on the original medium (e.g.,

digital tape) may exceed the rate required to depict the highest frequency of interest. For example, the sample rate for the data might be 2048, while the user is only interested in frequency components of 100 Hz or less. If it is assumed that about five data points per cycle are adequate to describe the 100-Hz frequency component, then 512 would be an adequate sample rate.

The File Creation Program provides a capability to change a sample rate that a user feels is excessive or that does not match the sample rate for a related item. The methods for sample rate change are described in Paragraph 3.2.3.

The user may need to use the rate reduction capability to make the sampling rate identical for all item codes of the same kind stored in a single partition. For example, certain absolute pressure sensor data may be recorded on digital tape using 2048 samples/sec, while other absolute pressure sensor data are recorded at 4096 samples/sec. By applying a skip rate of two to the data that have been sampled at 4096, all of the absolute pressure sensor data stored on the partition would be sampled at the same rate.

The highest frequency that can be represented by digitized data is called the Nyquist frequency. This frequency is one-half the sample rate. In reducing the sample rate, the user must consider the possibility of frequency aliasing. Suppose in the first example above that there was a significant noise component at 450 Hz. When the sample rate is reduced from 2048 to 512 the 450-Hz noise component would be aliased to appear at 62 Hz. The File Creation Program provides the capability to low-pass filter the data while transferring it to the Master File. In the pass-band region, the filter used is very flat in magnitude and does not distort the phase (see Figures 28 and 29).

Figure 28 shows the magnitude part of the 'transfer function' of the low-pass digital filter used in the File Creation Program when a breakpoint of 200 Hz is specified. This means that a Fourier component in the input time history would be multiplied by the corresponding magnitude part of the transfer function on output. The phase of the output would be undistorted for frequency components at or below the breakpoint of the filter (see Figure 29).

When the breakpoint frequency of the filter is changed, the scale of Figure 28 would have to be changed accordingly. The magnitude response can be thought of in terms of percent of the breakpoint frequency and percent of modulus transfer. Between 0 and 100 percent of the breakpoint frequency, the magnitude response is 100 ±1 percent. Past the breakpoint,

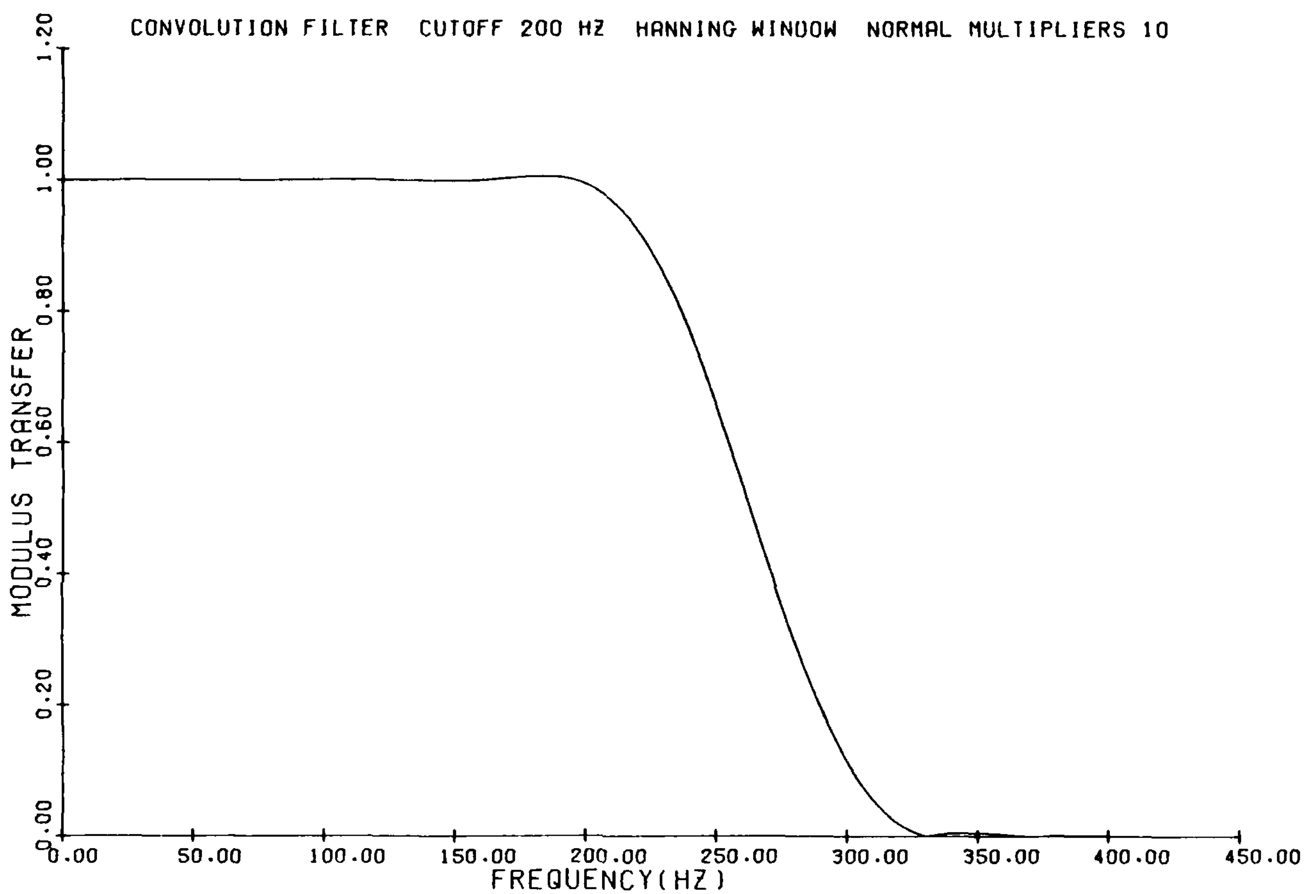


Figure 28. Convolution filter modulus transfer function.

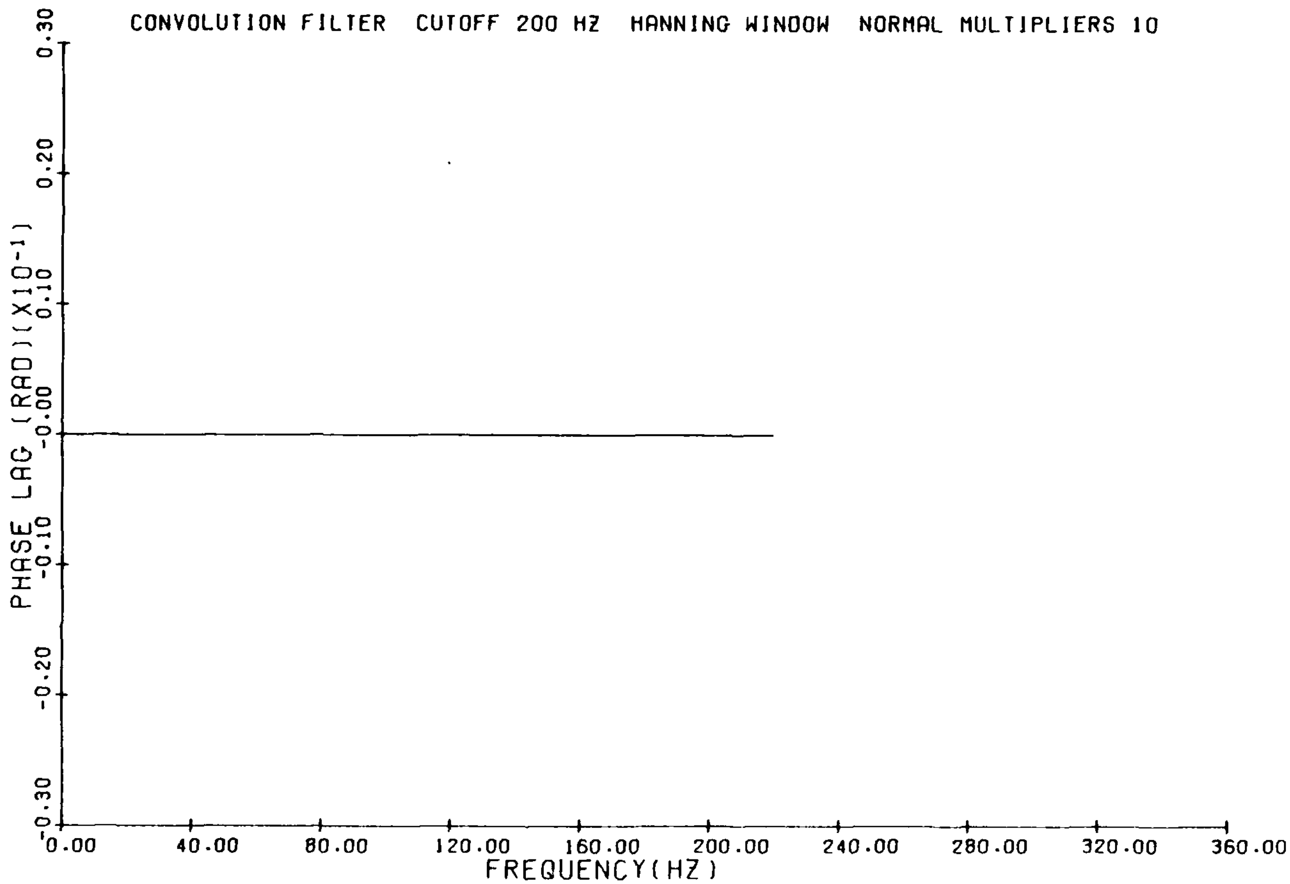


Figure 29. Convolution filter phase lag.

the response rolls off sharply to about 2 percent at 155 percent of the breakpoint frequency, and nearly 0 at 180 percent of the breakpoint.

The performance of this filter begins to deteriorate as the breakpoint is set below 1/80th of the original sample rate. Thus, data digitized with a sample rate of 4096 should not be filtered with a breakpoint much less than 50 Hz. This restriction applies only to the digital filter used in the File Creation Program.

When the user selects a sample rate reduction factor, aliasing may be prevented by selecting a filter breakpoint sufficiently low so that noise components with frequencies at or above the new Nyquist point are adequately attentuated. The filtering capability can also be used to create smooth data, which can improve the readability of plots. However, frequency components important to the data must not be attenuated significantly.

Those users who specify filtering for their data should be aware of the problem of edge effects or filter initialization. When generating filtered values near an edge of the specified output record, the filter must include in its calculations input data points corresponding to times outside of this edge. If such data are not present, erroneous filtered output can be generated for those values near the edge. If data outside the output record boundary are present on the data tape (if an offset is specified for example), that data will be accessed by the program for the filter to use. If such data are not available, a warning message is generated in the printout from the File Creation Program.

Operation of the Processing Program can be optimized for measured data input by reducing the sampling rate for the True Airspeed, OAT, and Boom System static pressure and filtering with as low a breakpoint as possible. However, the rotor azimuth data should not be filtered nor should its sample rate be reduced.

3.2.5 Time Alignment

For the OLS application of this system, an option is provided in the File Creation Program to correct for various time misalignments between the data from different sensors. Specification of the ALIGN command in the File Creation Program input will cause the program to read a tape containing a correction offset for each item code/counter pair. The correction offset will then cause the program to discard a certain number of points at the start of the data stream in the same fashion as an OFFSET. If an ABSOLUTE offset is specified for a counter, all time alignment offsets are ignored.

WARNING
When OLS data are being processed, the ALIGN option should always be selected unless the user has specific information that alignment offsets are invalid.

3.2.6 Info File Creation

The DTF format will accommodate the information necessary to specify Info File "groups" (See Section 5.8). This information specifies item codes, corresponding geometric positions, and collective labels for groups of like channels of data. The FCP will take any such information found and write it to a file in proper Info File format. More information must be joined or concatenated with this file to form a complete Info File. However, this capability should greatly simplify the effort required to create Info Files for simulated data where structural element positions may be adjusted frequently.

3.3 QUESTION AND ANSWER PROGRAM TO CREATE USER INPUT DATA SETS

This special program is provided for users to develop the control input for the File Creation Program in an interactive mode. Questions are asked to prompt input for every possible specification. The output of the program is a card image data set that can then be specified as the user input for the File Creation Program. This program will assure the user that his input for the File Creation Program is correct in syntax and that all possible specifications have been considered. This program cannot assure that the content of the generated input stream is correct. For example, this program cannot assure that a specified item code actually corresponds to the data the user wants.

Upon entry to this program, the user is guided step by step on a question-and-answer basis through all possible inputs. Although knowledge of the syntax for the File Creation Program is unnecessary, the user must be familiar with the effect of the various inputs (for example the user must know that OFF-SETs apply to all the item codes for a given counter). All of the necessary information regarding both the data to be transferred and the Master File partition must have been gathered before the user executes this program.

3.4 FILE MAINTENANCE

When the Master File is accessed by many users, there should be a designated data base monitor. This monitor must perform three functions. First the information on the Master File must be protected against some catastrophic loss of data (e.g., a disc hardware failure). Second, the storage requirements of the users must be coordinated. Third, to assure the availability of adequate storage space, the data present on the Master File must be scanned to assure that unused partitions are deleted. In the event that there were only a very few users in good communication with each other, these users might share the monitor function.

The File Maintenance Program is provided for use by the monitor in fulfilling the requirements listed above. The functions provided by this program are listed in Table 4 along with the commands that access those functions. One command and one qualifier is allowed per line of input to the program and the program executes the commands in the sequence in which they are read.

The partition map is the same as the map that can be written by the File Creation Program after data transfer is complete. The Master File MAP command lists each partition name along with offset to the first record, size in records, user, creation date, last access, and password. The Master File password (SUPERWORD) must have been entered before a Master File map can be generated that lists partition passwords. Otherwise, the partition passwords are left off the map.

The partition SAVE command copies a partition to digital tape for backup or archival use of the data. A Master File SAVE command copies the entire Master File to digital tape. The user must take care in using either SAVE that the partition records will fit on one digital tape. Currently, one Master File record contains 1024 bytes and the user must determine from the number of records in the partition or in the Master File, from the tape density and blocking factor, and from the tape size whether the data will fit. Only one SAVE (partition or Master File) can be performed per run of the File Maintenance Program.

The partition RESTORE command copies a SAVE'd partition from digital tape back to the Master File. The partition name specified with RESTORE cannot already be present on the Master File (see DELETE below) and there must be sufficient free storage space to accommodate the partition size.

The Master File RESTORE ALL copies a SAVE'd Master File image from digital tape back to the Master File disc area. The

TABLE 4. FUNCTIONS AND COMMANDS FOR THE FILE MAINTENANCE PROGRAM

FUNCTION	COMMAND
Map of a partition content	MAP (PARTITION NAME)
Map of partitions in the Master File	MAP ALL
Save on digital tape of a partition	SAVE (PARTITION NAME)
Save on digital tape of the Master File	SAVE ALL
Copy a partition from digital tape to the Master File	RESTORE (PARTITION NAME)
Copy a Master File image (SAVE) from digital tape to empty Master File	RESTORE ALL
Delete a partition from the Master File	DELETE (PARTITION NAME)
Enter a partition password	PASS (PASSWORD)
Enter password for the Master File	SUPER (SUPERWORD)
End processing	END

Master File disc area to be written on must be initialized, contain no partitions, and be sufficiently large to hold all the Master File records on the tape.

The partition DELETE command removes a partition from the Master File. Either the SUPERWORD or the partition PASSWORD must have been specified to DELETE a partition.

The SUPERWORD is a special password provided for the use of the Master File monitor. This password is defined during the Master File initialization run. The function of the SUPERWORD is to allow the monitor to DELETE partitions without knowing the individual password for each partition.

The File Maintenance Program can be run either interactively or in batch. The user should be aware that in interactive mode the MAP's will be printed on the terminal being used. No other program that assesses the Master File may run while the File Maintenance Program is executing.

4. PROCESSING PROGRAM - GENERAL ASPECTS OF USER INPUT

4.1 GENERAL USER INPUT RULES

User instructions to the Processing Program follow the same general free-field rules as the input to the File Creation Program. However, the structure and meaning of the commands are different. Commands are typed in as sequences of entries. Two entries are separated by one or more blanks, a comma, or a slash. The slash is a special separator that is explained later. An entry is a sequence of characters that does not include a separator. There are three kinds of entries: numbers, strings, and nulls.

A number begins with a digit, decimal point, plus sign, or minus sign. Subsequent characters can include digits and/or a decimal point. Only one decimal point is allowed in a numeric entry. Exponential (power of ten) numbers are not allowed. Decimal points are not required for floating numbers if no fractional part is included. Following are some examples of valid and invalid numeric input:

Valid	Invalid
1024	.126.4
1024.	10.5E-6
+10.2458	128A5
.1024	-10.4-3

A string entry begins with any character other than a separator, digit, plus sign, minus sign, or decimal point. Subsequent characters in a string can be anything except separators. Alternatively, a string may be enclosed in single quotes (i.e., apostrophes). When a string is enclosed in single quotes, it may start with and contain any characters except the single quote. In fact, the single quote should never be used except to enclose strings. Following are examples of valid and invalid string entries:

Valid	Invalid	Reason Invalid
P823	8PAT	Starts as number
CONTOUR	IN,DY	Contains separator
AZY+*!	+ABC	Starts as number
'8PAT'	'CAN'T'	Contains single quote
'3RD POSITION'	CAN'T	Contains single quote

A null entry is indicated by two commas in sequence. In some cases, null entries are used to imply default entries (see Section 4.4).

A line consists of a sequence of entries contained in character positions 1 through 55. The particular character position where an entry begins is not important in evaluting a number or identifying a string. The sequence of entries is vital. An entry cannot extend from the end of one line to the beginning of the next.

When string entries are to be identified for meaning, the characters are matched with a list of model character sequences. The characters are matched character by character for the first four characters. If fewer than four characters are input, then the match can still be made so long as it is unambiguous. Thus, the model entry for HARMONIC is HARM. However, HAR is an adequate match for HARM since it is unambiguous. Some more examples are:

DER = DERI = DERIVE
DIS = DISP = DISPLAY
ED = EDIT
EXE = EXEC = EXECUTE

Extra characters in an entry up to the fourth character must match. Thus,

DERX ≠ DERI
EDZZ ≠ EDIT

4.2 COMMAND STEPS AND SUBSTEPS

Commands are input and executed in steps. A step is a self-sufficient action by the program in that the computer can proceed without intermediate input from the user (exceptions to the rule are CHANGE mode of EDIT, printout control, MENU listing control, and cursor control for plots) and some tangible result is produced - either an output, an intermediate storage of data, or a change in state of the program run. No subset of a complete step will produce tangible results from the computer. In fact, a step that is incompletely entered can be erased at any time with a 'CANCEL' control command and the computer is returned to the status held before starting the step.

A step will include one to four substeps, which fall into four categories: Specification, Action, Input, Disposition. A Specification is always required. The other substeps may or may not be needed depending on the nature of the Specification and/or Action substeps. Table 5 is a list of the specifications, which are all single entries, along with associated required substeps. The parentheses for the Input Substep following the UTILITY Specification means the presence of an Input Substep depends upon the Action Substep entry specified.

TABLE 5. REQUIRED SUBSTEPS FOR EACH SPECIFICATION

SPECIFICATION	ACTION	INPUT	DISPOSITION
ANALYZE	X	X	X
DERIVE	X	X	X
DISPLAY		X	X
EDIT		X	
BUILD		X	
SAVE			
NOEDIT			
EXECUTE		X	
MENU		X	
TERMINATE			
COMMENT		X	
SET	X		
UTILITY	X	(X)	

If a substep is not required for a particular specification, the input routines will assume it is not present. Thus, substep input following the Specification Substep DISPLAY will be assumed to belong to the Input Substep rather than an unnecessary Action Substep.

The Specification Substep indicates the sort of process to be performed in the step. Sometimes, the Specification Substep entry does not require amplification in the form of additional substeps (e.g., 'TERMINATE'). Usually, additional information is required to define the step so that one or more of the other substeps are required.

Examples of actions would be a filtering process or derivation of C_n. The Action Substep is required to define the Specification Substeps more precisely.

The Input Substep specifies where the data for processing and display are obtained. It can specify the number of dimensions of the input and independent variables to be used. Specialized entries are required for such specifications as 'EDIT' and 'COMMENT'.

The Disposition Substep describes the form of the output of the step (e.g., contour plot, save on scratch disc file, etc.).

Substeps are separated by slashes. Recall that a slash is a special separator so that a slash may be placed between the last entry of a substep and the first entry of the next substep without need for other separators to divide the two entries. More than one substep can occur on a line, or a substep can take more than one line to input. Conceivably, an entire complex step could be entered on one line. A slash is required after the last substep to begin execution of a step.

4.3 DEFAULTS

The use of default entries can be of value to the user for two reasons. First, a default is a quicker entry, saving the user time. Second, a default requires fewer character inputs, which reduces the possibility of error during input. In using defaults, however, the user bears the responsibility of knowing what the defaults are and how the default specifications and structure work. Defaults can be identified by use of the HELP entry (see Paragraph 4.5.1).

Default values and options are specified through more than one procedure. Some parameters have defaults that are permanently specified in the program. Other parameters retain the value last specified during the run. Sometimes, a parameter is set to an initial value at the start of a run or session and then changed as nondefault values are specified. For other parameters, no default exists until a value has been specified by the user once; an attempt to default to the uninitialized value produces an error message. (For example, there is no default counter as a user session begins. Once a counter is specified, that counter is the default until a new one is specified.)

Both single entries and groups of entries can be defaulted. The simplest default for an entry occurs when that entry is the last of a substep. The entry is simply omitted, and the slash marking the substep end is inserted immediately. For example, instead of:

```
FILTER 12 3 4/...
```

the user could leave off the last entry since "4" is the default number of poles for digital filtering:

```
FILTER 12 3/...
```

A group of entries at the end of a substep can be defaulted in the same fashion. Thus:

```
CP, 264, CALC, CALC/...
```

could be entered

CP/...

if the rotor radius of 264 inches had previously been entered.

If an entry to be defaulted occurs before an entry in the same substep that must be entered, then the default can be specified by the presence of two comma separators in sequence. For example,

MCQ,5,,80,13/...

would cause the third entry, rotor radius, to default to a previously entered value. More than one default in sequence can be specified by more than two commas. For example,

MCT,5,,,80,13/...

would cause the third entry, ship gross weight, and the fourth entry, rotor radius, to default to previously entered values.

4.4 ERROR HANDLING

User-input errors detected by the program fall into two major categories, invalid line and invalid entry, depending upon the effect of the error on the input sequence.

Invalid line errors consist of three types. The first is numeric entries that cannot be successfully interpreted as numbers by the free-field input routine. Such errors include embedded illegal characters in an entry that begins with a numeric entry character (+, -, ., 0-9), and numeric entries with multiple decimal points. The second type of invalid line has characters typed beyond column 55. The third type of invalid line occurs when a single quote cannot be matched to another single quote to terminate a string (i.e., when an odd number of single quotes appear on a line).

An invalid line error will cause the program to output a line containing only one character, a '+', pointing to the position where the error was detected. In the interactive mode, the user must then retype the entire line correctly and proceed as before. In batch mode, the program terminates without execution after scanning the remaining lines of input for error.

An invalid entry error is essentially any other error that can be detected in the user interface. When such an error is detected in an entry, that entry is invalidated and all subsequent entries on the same line from the same or subsequent

substeps are invalidated. Previous entries on the same line remain valid. The program prints a message indicating the erroneous entry, and converts to HELP mode for the remainder of the substep (see Section 4.5).

Not all errors can be detected in the user interface part of the program. A command step may have proper syntax and structure but be impossible to execute. For example, the user might request a True Airspeed derivation although data from the airspeed sensor is not available for processing. When errors such as these are encountered in processing, an error notification and number is output. Frequently, a diagnostic is output. Occasionally, an error is only identified by an error number. These numbers are identified in Appendix A.

4.5 CONTROL COMMANDS

Certain string entries are recognizable throughout the specification of a step. These control commands do not constitute values for the entry positions but instead require some instant action on the part of the input routines. Subsequent entries on the same line are ignored by the input routines. The control commands are HELP, LIST, and CANCEL.

4.5.1 HELP Command

HELP, or the alias '?', will cause the program to prompt the user for proper input. If HELP or '?' is substituted for any entry of a substep, including the first, the user will be instructed as to the meaning of each remaining entry and, where possible, which allowed options are available for each entry of that substep. Sometimes the user will be asked for one entry at a time in the HELP mode, since each entry can affect the meaning of subsequent entries in the substep. On other occasions, entire sequences of entries will be requested from the user. The default for an entry, if any, is specified in the HELP message.

A specific example will now be considered. An exchange between the computer and the user making maximum use of HELP is shown and then explained line by line. User entries are underlined while computer outputs are not. Line numbers are included for reference only and would not appear in an actual user/computer interchange.

1. NEW STEP.

2. HELP

3. SPECIFICATION SUBSTEP

```
 ENTER:- (1)OPTIONS: ANALYZE,DERIVE,DISPLAY,BUILD,
                ,EDIT,NOEDIT,EXECUTE,MENU,TERMINATE,SAVE,
                COMMENT,SET,UTILITY  -NO DEFAULT
DISPLAY
 SPECIFICATION SUBSTEP  COMPLETE  /
 INPUT SUBSTEP
?
 INPUT SUBSTEP
 ENTER:- (1)INPUT SOURCE: SCF1,SCF2,SCF3,GROUP,(ITEM
                CODE) -NO DEFAULT
P157
 ENTER:- (1)(COUNTER)-NO DEFAULT
         (2)TIME(SEC)-NO DEFAULT
         (3)DURATION(SEC)-NO DEFAULT
7 B
1Ø65 Ø 1
 INPUT SUBSTEP  COMPLETE  /
 DISPOSITION SUBSTEP
?
 DISPOSITION SUBSTEP
 ENTER:- (1)OPTIONS: PLOT,MPLOT,LPLOT,DPLOT,APLOT,
                PRINT,CONTOUR,SURFACE,KEEP,ADD -NO DEFAULT
PLOT
 ENTER:- (1)IND AXIS OPTIONS: TIME,FREQ,HARM,ROW,
                COLUMN,MRAZ,MRPM,TAS,IMPL - DEFAULT-'IMPL'
         (2)CURSOR OR CLOSE-DEFAULT-'CLOS'
```

```
29.           (3)INTERVAL Y: AUTO,LOG,(INTERVAL)-
30.                DEFAULT-'AUTO'
31. TIME CLOSE AUTO
32.  ENTER:- (1)BOTTOM Y: AUTO,(Y)  -DEFAULT-'AUTO'
33.           (2)INTERVAL X: AUTO,LOG,(INTERVAL)-
34.                DEFAULT-'AUTO'
35. AUTO AUTO
36.  ENTER:- (1)MINIMUM X: AUTO,(X) -DEFAULT-'AUTO'
37. AUTO
38.  DISPOSITION SUBSTEP  COMPLETE
39.  WAITING FOR SLASH TO EXECUTE STEP.
40. /
41.  EXECUTING
```

In line (1), the program informs the user that the previous step has completed or aborted and a new step is beginning. Line (2) informs the computer that the user does not know what the available Specification Substep entries are. The computer responds with line (3), which informs the user that input will be for the Specification Substep, and with lines (4) through (6), which list the entry options. The user selects DISPLAY on line (7). On line (8) the computer indicates that the Specification Substep is complete. A slash is included to substitute for the slash the user would have entered if HELP were not active.

Since the program exited the Specification Substep in the HELP mode, the user is told the new substep name in line (9) and the HELP mode is terminated. The user still cannot remember what to do, however, and so requests more HELP with line (10) using the '?' alias. In line (11), the computer reinforms the user that the current substep is 'INPUT' and lists possible first entries on lines (12) and (13). The user responds with an item code entry in line (14).

In lines (15),through (17), three different entries are prompted at the same time. The computer requests the counter, the time offset from the start of the time history, and the time duration of data to use. In line (18), the user responds

by entering counter 1065, time offset zero, and record length one second. Since all entries for the Input Substep are now specified, the user is advised that the substep is complete in line (19) and a slash is supplied by the computer. Then, in line (20), the computer informs the user that the new substep is 'DISPOSITION'.

The user requests additional HELP in line (21), and the computer responds in line (22) that the current substep is 'DISPOSITION'. In lines (23) and (24), the output options are listed for the first entry of the Disposition Substep. The user selects a simple X-Y plot in line (25). In lines (26) through (30), the user is requested to define the independent axis variable, the status of the Tektronix crosshair cursor following generation of the plot, and the scaling interval for the dependent plot axis. In line (31), the user selects TIME as the independent variable axis, no cursor activation, and automatic selection of the Y (or dependent variable) axis scale interval. The program then requests the minimum 'Y' axis value to depict and the scaling interval for the 'X' (or independent variable) axis in lines (32) through (34). Automatic scaling is chosen by the user for both cases in line (35). The program then requests the minimum 'X' value to depict in line (36) and the user again selects automatic scaling in line (37).

The Disposition Substep is now complete and the computer announces this fact in line (38) without inserting a slash to terminate the substep. In line (39), the computer informs the user that a slash must be entered to execute the step. The user provides the slash in line (40), and the computer responds in line (41) that the step is being executed.

These 41 lines used to define one command step may appear to be unduly time-consuming. Notice, however, that if the HELP mode is not invoked, the above 41-line transaction may be accomplished by the following three lines:

1. NEW STEP.

2. <u>DISP/A938 676 0 1/ PLOT TIME CLOS AUTO AUTO AUTO AUTO/</u>

3. EXECUTING

By using defaults, line (2) above may be further reduced to:

2. <u>DISP/A938 676 0 1/PLOT/</u>

Some important features of the HELP format can be observed in the example given. The HELP messages are indented at least one column position. In fact all program output is indented

at least one column position. User input can begin in column one. In the interactive or interactive-graphics mode, this feature should distinguish user inputs and program messages. In addition, the default for each entry is listed in the HELP message. If there is no default for an entry, then this fact is indicated.

A HELP message specifically requesting one or more entries will always begin with 'ENTER:-'. More than one entry can be prompted in one message and the individual entry prompting messages are identified with a number enclosed in parentheses. This number refers to the entry position that the program is currently waiting for rather than the entry number in the substep. Thus, a "(1)" refers to the first entry that the user must type. Entry options are separated by commas while separate entry prompting messages begin on separate lines.

HELP messages indicate numeric value options and nonkeyword string options by enclosing either a descriptor or the corresponding units in parentheses. Keyword string options are not enclosed in parentheses. For example, in line 33 above, "AUTO" is a keyword and is not enclosed in parentheses, while "INTERVAL" is a numeric value descriptor and is enclosed in parentheses.

The HELP command can be used in conjunction with defaults. As before, commas can be used in sequence to specify default entries for a substep. For example, line (31) above could be entered:

31. TIME, AUTO

since CLOSE is the default response for the Tektronix cursor activation status. The slash can also be used to specify defaults for the last entries in a substep. Line (31) could be entered:

31. TIME/

32. EXECUTING

where all the remaining DISPOSITION substep entries are set to defaults and execution of the step begins immediately.

HELP mode is immediately terminated when the slash is entered. The user then has the option to specify entries for subsequent substeps on the same line. For example, line (18) could be entered:

AD-A095 188 BELL HELICOPTER TEXTRON FORT WORTH TX F/G 9/2

THE DATA FROM AEROMECHANICS TEST AND ANALYTICS -- MANAGEMENT AN--ETC(U)

DEC 80 R B PHILBRICK DAAK51-79-C-0015

UNCLASSIFIED BHT-699-099-025-VOL-1 USAAVRADCOM-TR-80-D-30A NL

2 OF 3

AD A095188

18. <u>1065 0 1/PLOT/</u>

19. EXECUTING

In fact, the user always has the option to specify additional entries that follow those actually prompted by a HELP message.

4.5.2 LIST Command

If HELP or defaults have been used in generating a command step, the user may become uncertain about the exact entries used. By specifying LIST, the user can obtain a complete listing of the entries already specified in the step input, including any defaults. LIST can be entered at any time prior to entry of the slash terminating the final substep. The listing will include all entries specified for the step up to the entry just prior to the position occupied by the LIST command. However, if LIST is entered before any entries are made for a step, the previous step will be listed. This feature can be of considerable use if a step execution has gone wrong and the user wants to determine whether the proper command entries were made.

After a LIST is completed, the program will expect input entries beginning with the entry position for which LIST was substituted. If the program was not in the HELP mode when the user entered the LIST command, HELP will not be active when the LIST is complete. If HELP was active when the LIST command was entered, then HELP will still be active when the LIST is complete and the user will be reprompted for entries beginning with the entry the LIST command replaced. If the user entered LIST after typing all the entries for the final substep but before the final slash, the program will still be waiting for that slash after the LIST is complete.

4.5.3 CANCEL Command

By entering CANCEL, all input entered for the step to that point is canceled and the program returns to the beginning of a new step.

4.6 COMMAND SEQUENCING (Edit)

The Edit or Command Sequencing function allows the user to create, modify, execute, or delete sequences of command steps. These sequences are stored on a permanent disc file so that sequences created in one run of the Processing Program can be executed in other runs of this program. In these separate runs, the operating mode (i.e., Batch, Interactive, Interactive Graphics) may be different. Following are some

possible reasons for using the Edit or Command Sequence capability:

- The user may want to enter instructions interactively using the error checking process, but without execution of steps, and then execute the identical instructions in batch mode.

- The user may want to enter instructions interactively and see the results immediately and then get incremental plotter and line printer versions of printout and plots from batch without retyping the instructions.

- The user may have long sequences of instructions to enter that are generally the same for all given executions but that involve a small number of changes.

Commonly, the user may have a combination of these reasons for using the Edit process.

This process involves creation, modification, and deletion of blocks of instructions. A block is a sequence of one or more steps of user instructions and is limited to no more than 112 lines of 55 characters, regardless of the number of command steps included. Each block has a four-character block name.

A block of instructions can be used by entering EXECUTE as the Specification Substep and the block name in the Input Substep. The instruction lines are extracted from discs in sequence just as though the user is entering them. When the program encounters a NOEDIT Specification Substep in this sequence, it exits the EDIT block execution mode and reenters the direct mode. If a command input error is encountered while in block execution mode, the program returns an error message and exits this mode.

As many as four parameters may be passed to a command sequence block in an EXECUTE command. A parameter may be a string or a number. Parameters are identified by entry position following the block name in the EXECUTE command. Each parameter must be preceded by the character "%". Parameters are inserted in the command sequence block whenever a "%" is encountered in a line of the block. A digit (1, 2, 3, or 4) follows the "%" character in the block to identify the parameter to insert. For example, the second parameter following the block name in the EXECUTE command would replace the characters "%2" wherever they occur in the command sequence before the corresponding lines are read by the Processing Program. This capability allows a command sequence to be used repeatedly for many different values of a parameter. For example, the same

command sequence could be executed several times in a run for different counters by supplying the counter as a parameter for each execution of the command sequence.

Blocks of instructions can be created by using the EDIT specification with NEW entered in the INPUT substep or by using the BUILD specification. If EDIT with NEW is specified, then the program enters a mode such that commands are read with normal error checking. However, when the final slash is entered requesting execution of the step, then all the lines comprising the step are copied to the Edit block on disc and the step is not executed. Note that error checking is done on input syntax or unreasonable or unrecognizable entries but not on errors that would be detected in executing the step. Thus, if the user requests a PLOT of processed data that is not stored on the scratch disc file specified for input, the error would be undetected. The user is free to use HELP, CANCEL and LIST while in the EDIT mode as control commands will not appear in the command chain created.

Exit from this mode is accomplished with a NOEDIT Specification Substep. At that point, a NOEDIT Specification Substep is written for the last line of the Edit block. The block is then available with the assigned name in the Edit file.

The second way to create Edit blocks is by using the BUILD Specification. An Edit block is started and the program returns to direct mode. The program actually executes each step when the final slash is entered. The user can then record the lines of instruction comprising the previously executed step by specifying SAVE as the next step. An intervening LIST control command will not interfere with the SAVE. The user completes the Edit block by entering NOEDIT as in the EDIT/NEW method.

The BUILD and SAVE specifications allow the interactive graphics user to actually try out the execution of steps and save only the commands that produce desired output. Data can be plotted on the Tektronix, scanned for correctness, and then the same instructions can be executed in batch to produce Calcomp or DP-1 plots. Of course, the user must also SAVE intermediate steps that process and store data. SAVE'd processing instructions that later appear superfluous or incorrect can be removed with the CHANGE mode of EDIT.

Creation of Edit blocks with the EDIT/NEW mode provides advantages different from those provided by the BUILD mode. First, commands that create plots can be entered on a non-graphics terminal. Second, commands are entered with syntax checking but without execution so that considerable time can be saved in generating a command sequence block.

Two other modes of EDIT are available: CHANGE and DELETE. DELETE is quite simple. The Edit block name is removed so that the space and the name are available for later use. The CHANGE mode is more complicated and is not envisioned for use by beginning users. CHANGE allows the user to alter an existing Edit block. The user must be cautious in the change mode since line modifications are checked only for line errors (see Section 4.1) and syntax checking is not performed on the resultant instruction sequence until the Edit block is actually executed. Paragraph 5.7.1 provides additional information on the CHANGE mode. The CHANGE mode of EDIT must be used to condition Edit blocks to receive the parameters passed in an EXECUTE command.

4.7 'INFO' FILES

During execution, the program must frequently access certain item codes for derivations (e.g., the rotor azimuth item code). The program will also need to access groups of item codes that measure like parameters along with geometric positions and matrix (row, column) organization for these item codes. If the user was required to enter all this information for every command step requiring it, the process would prove time-consuming, tedious, and prone to error.

The 'Info File' is provided to minimize such repetitive entries. This file consists of two parts. The first part is a short list of keywords with corresponding critical item codes such as rotor azimuth and indicated airspeed. In addition, the first part may include specifications for unit conversion of dependent variables on output from the program. The second part consists of several groups of item codes of like kind. Each group has a four-character name and includes labeling information, row (chord) and column (radius) positions, and the corresponding item codes for the row-column intersections. A group can have multiple rows and columns (a two-dimensional group), a single row and several column positions or a single column and several row positions (one-dimensional groups). Groups can specify both double-row elements or only a single double-row element.

The file format is sequential and consists of a number of 70-column lines, which can be maintained as such in card image form. For the OLS application, a specific Info File is provided. However, a user may create special Info Files for different applications containing different item codes and different geometric information. Rules for the structure of an Info File are contained in Section 5.8.

When specifying input data with the Info File, the user simply references the appropriate group by name. The MENU

command can be used to list the groups in the Info File by name and title. Entry options are provided for the user to select a single element or all elements in both the row and column directions. The user also has the option to select BOTH double-row elements or either the TOP or BOTTOM double-row element. For example, in the OLS application using the absolute pressure sensor group, S2PP, the user could select ALL for the row elements (chord positions), '3' for the column elements (75 percent radius position), and BOTTOM for the lower surface sensors.

Each group can specify an individual azimuth correction angle for the sensors in the group. The program uses this angle to align azimuth when sensors are mounted on different rotor blades. For the OLS blade application involving a test with a two-bladed helicopter, azimuth was measured for the red blade, but all of the blade absolute pressure sensors were mounted on the white blade. The azimuth correction angle of 180 degrees that is contained in the group that specifies these sensors (S2PP) is used to shift azimuth appropriately.

5. SPECIFIC PROCESSING PROGRAM USER INSTRUCTIONS

5.1 PROGRAM INITIALIZATION PHASE

Immediately upon the beginning of program execution and before the command steps described in Sections 5.2 through 5.6 can be accepted, the computer executes a Program Initialization Phase. Two types of activity occur during the Initialization Phase: (1) the program elicits certain information from the user regarding the run; and (2) the program initializes or verifies certain disc files. This phase occurs in all three operating modes: Batch, Interactive, and Interactive Graphics. In the Batch mode, the user must assure that the proper entries for this phase are present at the beginning of the user input sequence. When the program is running in the Batch mode, an erroneous input for the initialization phase terminates program execution.

The very first information requested by the Processing Program is the mode of operation. The program prints the following message:

```
BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) - PROCESSING PROGRAM

ENTER OPERATING MODE:
  1  =  BATCH
  2  =  INTERACTIVE (NO PLOTS)
  3  =  INTERACTIVE GRAPHICS (TEKTRONIX NEEDED)
```

The user should enter the number corresponding to the proper operating mode. The program will repeat the selected number back to the user (in the Batch mode, only the echoed number will appear in the printout).

For any of the inputs to the Program Initialization Phase, the user may substitute the alternative keywords 'RESTART' or 'END'. 'RESTART' will cause the program to return to the beginning of the initialization sequence where the message listed above is printed. The user will then have the opportunity to repeat the entire sequence of Initialization Phase entries. 'END' will cause the program to terminate.

Following the entry of a number to specify the operating mode, the program prints a list of operating settings and keywords for modification of these settings. This list is shown in Figure 30. The user is then prompted to enter 'YES' to accept the existing settings or one of the listed keywords to

```
BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) - PROCESSING PROGRAM

 ENTER OPERATING MODE:
   1 = BATCH
   2 = INTERACTIVE (NO PLOTS)
   3 = INTERACTIVE GRAPHICS (TEKTRONIX NEEDED)
3
3

      RUN SETTINGS:                                  KEYWORD
  TERMINAL DATA RATE 240 CHARACTERS/SECOND            'LINE'
  ROTOR MODE 'MAIN'                               'MAIN','TAIL'
  PLOT GRID MODE  'GRID'                         'GRID','NOGRID'
  PLOT TICS MODE 'NOTICS'                        'TICS','NOTICS'
  PLOT FRAME WIDTH  8.50 INCHES                       'PWID'
  OPERATOR PEN PLACEMENT IN 'X' -1.50 INCHES          'PENX'
  OPERATOR PEN PLACEMENT IN 'Y'  0.50 INCHES          'PENY'
  PRINT BLOCKS OF 5 LINES/BLOCK   6/PAGE             'BLOCKS'
  SCRATCH FILES SIZE (EACH)  225 RECORDS            'FILESIZE'
  SCRATCH FILES ARE PERMANENT                     'TEMP','PERM'
  CPU SECONDS TO TRIGGER WARNING 90.00                'WARN'
  STEP EXECUTION TIMES WILL NOT BE PRINTED       'STEP','NOSTEP'

  ENTER 'YES' TO ACCEPT THESE VALUES OR
  A KEYWORD TO MODIFY A SETTING.
STEP
  STEP EXECUTION TIMES   WILL   BE PRINTED       'STEP','NOSTEP'

  ENTER 'YES' TO ACCEPT THESE VALUES OR
  A KEYWORD TO MODIFY A SETTING.
YES
 INITIALIZING SCRATCH FILES.  PLEASE WAIT.
 ENTER PARTITION NAME
C81COMPR

 NEW STEP  STEP CPU  0.20 SEC  TOTAL CPU   0.2 SEC
```

Figure 30. Initialization phase run setting modification sequence (example).

modify a setting. When there is a selection between two keywords to indicate one setting, the keyword entry alone is sufficient to define the setting and modification. When a single keyword is available to select a setting, then, after this keyword is entered, the computer will prompt the user for a numeric entry. This number may be entered in free field format using the rules for numeric entries presented in Section 4.1. However, the number should not be entered on the same line as the keyword.

Some of these run settings can be changed during the program run as well as during the Initialization Phase. The settings that <u>cannot</u> be changed later are Terminal Data Rate, Plot Frame Width, Operator Pen Placement in 'X' or 'Y', Print Blocks, Scratch Files Size, Scratch Files Permanent or Temporary, CPU Seconds to Trigger Warning, and Step Execution Times Printing.

Figure 30 shows an example of modification of these settings. The user enters 'STEP' to specify that the CPU time required to execute each step will be listed at the beginning of the following step. No numerical input is required to modify this setting. The computer immediately lists the modified setting and then prompts the user to enter 'YES' to accept the new settings or a new keyword to further modify the run settings. The user enters 'YES' to terminate action on the run settings and the computer prints the message: 'INITIALIZING SCRATCH'. This message is informative and no input from the user is required at this time.

The following list gives the meaning for each of the user selectable run settings listed in Figure 30.

<u>TERMINAL DATA RATE</u> - This entry is significant only when the Interactive Graphics mode is used with a Tektronix terminal. The number entered should give the data communication rate between the terminal and the computer interface in characters per second. For example, if the data communications rate were 2400 baud, then the proper number for this entry is 240 characters per second, which is the default value listed in Figure 30. Normally this default is set for the proper rate at a given installation. The keyword to modify this setting is 'LINE'.

ROTOR MODE - This entry allows the user to specify which rotor will be used to specify rotor azimuth and rotor cycles for input specification, processing, and output scales. 'MAIN' specifies the main rotor, and 'TAIL' the tail rotor. 'MAIN' is the default rotor selection. The Info File relates the selected rotor mode to an item code (see Section 5.8).

PLOT GRID MODE - This entry instructs the computer whether to draw a grid on X-Y plots with linear scales. The keyword 'GRID' specifies that a grid line will be drawn for each annotated position on a linear scale. 'NOGRID' specifies that no grid lines will be drawn for linear scales. Grid lines will always be drawn for log scales regardless of 'GRID' or 'NOGRID' selection. 'GRID' is the default entry for this selection.

PLOT TIC MODE - This entry instructs the computer whether to draw tic marks on linear scales of X-Y plots. The keyword 'TICS' specifies that tics will be drawn while 'NOTICS' specifies that no tic marks will be drawn. This entry does not affect tic marks drawn for log scales. 'TICS' is the default entry for this selection in batch mode, while 'NOTICS' is the default for the interactive modes.

PLOT FRAME WIDTH - This entry controls the spacing between plot frames when an incremental plotter is used. The keyword 'PWID' specifies that the user will enter the horizontal width allowed for each frame of an incremental plot including the frame itself and the spacing between frames. This entry has no effect on the size of an actual plot frame. The value entered for the frame width must be between 7.5 and 20.0 inches. The default for this entry will vary from installation to installation.

OPERATOR PEN PLACEMENT IN 'X' - This entry informs the computer of the initial incremental plotter pen placement in the 'X' direction relative to the standard pen starting position. The standard pen starting position is one-half inch to the right and 3.5 inches down from the lower left-hand corner of the first plot, assuming that the first plot is an X-Y plot. For the Houston

Instruments DP-1, using preperforated paper, this <u>standard pen starting position</u> is one-half inch up from the lower perforation and one-half inch to the right of a perforation separating two sheets of paper. The operator pen placement specifications allow the user to adjust the plots for differences in initial placement of the pen. For the numbers used in Figure 30, the operator pen placement should be 1 inch above the lower perforations and 1 inch to the left of the perforations separating two pages. Allowed 'X' pen placement specifications are -7.5 to 7.5. The keyword for pen placement in the 'X' direction is 'PENX'. The default for this value will change from installation to installation.

<u>OPERATOR PEN PLACEMENT IN 'Y'</u> - This entry informs the computer of the initial incremental plotter pen placement in the 'Y' (vertical) direction relative to the <u>standard pen starting position</u>. See the previous entry description for a definition of this position. The allowed 'Y' pen placement specifications are -7.5 to 7.5. The keyword for pen placement in the 'Y' direction is 'PENY'. The default for this value will change from installation to installation.

<u>PRINT BLOCKS OF 5 LINES/BLOCK</u> - This entry tells the computer how many blocks to print on a page of output. A block consists of five lines of printed data and one blank line. Thus the total number of lines on a page is this entry times six plus a few header lines. The keyword for this entry is 'BLOCKS'. Allowed values are 1 to 1000. Defaults for this entry change depending upon the mode of operation.

<u>SCRATCH FILES SIZE</u> - This entry specifies the size, in records, of each of the scratch files, SCF1, SCF2, and SCF3. The specified size also applies to the temporary scratch file, which is otherwise invisible to the user. The keyword for this entry is 'FILESIZE'. Generally the default value for this specification will correspond to the size of the scratch area provided in the JCL so that the JCL must be modified to allow this number to be increased. If a number of records that exceeds the available disc space is specified, an abnormal termination of the program run will occur. Allowed values range

from 0 to 9999. The default for this entry will change from installation to installation.

SCRATCH FILES ARE PERMANENT/TEMPORARY - This entry specifies whether the scratch files must be initialized before they can be used. If the scratch files are TEMPORARY, they will be initialized. If the files are PERMANENT, they will not be initialized. Thus, newly allocated scratch files should be called TEMPORARY, whether they will be retained for repeated use or not. Files that have been used previously should be called PERMANENT to avoid reinitialization. Initialization of scratch files is a time-consuming process and will destroy any data previously stored on the files. For some installations of DATAMAP, it is possible for the system command language (e.g., Job Control Language for IBM installations) to send a message to the program that indicates whether the scratch files should be considered TEMPORARY or PERMANENT. If this capability is not available, the scratch files are assumed to be TEMPORARY and the user must override that setting with a PERMANENT entry if the scratch files need not be initialized.

8 CPU SECONDS TO TRIGGER WARNING - This entry specifies
F the number of Central Processor Unit (CPU) seconds which the program will use before generating time warning messages. When the specified time is exceeded, the warning messages will appear before each 'NEW STEP' message and list the CPU seconds used by the program up to that time. The keyword for this entry is 'WARN'. Allowed values for this entry are 0.001 through 999 seconds. The default for this entry changes from installation to installation.

STEP EXECUTION TIMES PRINTING - This entry instructs the computer whether to print the CPU execution time for each command step during the current run. These times are printed along with the 'NEW STEP' message that prompts the user for the subsequent step input. The keyword 'STEP' specifies that these CPU times will be printed while 'NOSTEP' specifies that the times will not be printed. The default for this entry is 'NOSTEP'.

When the message 'INITIALIZING SCRATCH' appears on the screen, the computer proceeds to initialize the scratch files SCF1, SCF2, SCF3 and the temporary scratch file. If the scratch files are PERMANENT, the computer will only test that the files are initialized and that the number of records specified are present. In addition, the Info File initial group (see Section 5.8) and the Command Sequence File are checked for validity. This process may require as much as a few minutes if the computer is extrememly busy or if the scratch files are exceedingly long. When the initialization is complete, the computer prints the following prompting message:

```
ENTER PARTITION NAME
```

The user should enter the name of the Master File partition containing the appropriate data for the run. When the corresponding partition is found, the computer will immediately display the prompting message, 'NEW STEP', and the Initialization Phase will be complete. If the partition name that has been entered is not found, then the computer will display the message

```
PARTITION 'JONES1' NOT FOUND
(ENTER 'MENU' FOR A LIST OF PARTITIONS)
ENTER PARTITION NAME
```

assuming that 'JONES1' was entered as the partition name. The user should now enter the corrected partition name or 'MENU' to obtain a list of the partition names. Figure 31 shows an example of the listing that is obtained with the 'MENU' entry.

5.2 SPECIFICATION SUBSTEP COMMANDS

The Specification Substep has a single entry position. There are 13 allowed keywords for this entry, which were listed in Table 5. The function of each of these keywords is given here.

ANALYZE and DERIVE each specify processing to be performed upon data. Data are accessed, processed, and output with either of these specifications. The actual process to be performed is identified in the Action substep (Section 5.3). An ANALYZE function performs some process on a general class of data (e.g., time histories). A DERIVE function performs a process on particular measured and/or derived parameters to produce a specific derived parameter output. DISPLAY has the same capabilities as ANALYZE and DERIVE for input and output. However, DISPLAY performs no processing function.

MENU

LIST OF PARTITIONS IN THE MASTER FILE

PARTITION	LENGTH	USER NAME	DATE CREATED	LAST CHANGED	LAST ACCESSED
D800RED	3607	R-PHILBRICK	09/23/78		04/19/80
D800WHIT	7097	R-PHILBRICK	09/23/78		06/21/80
PUSHOVER	315	R-PHILBRICK	01/17/79	07/19/79	04/19/80
C810LS35	2153	VANGAASBEEK	02/15/79		04/07/80
C810LS45	428	VANGAASBEEK	02/15/79		02/22/79
MANUDATA	115	VG	04/10/79	04/27/79	04/27/79
HQDMTEST	491	WALTERS	06/19/79		06/20/79
OLSBLACC	8720	VG	06/26/79		06/28/79
OLSHVFLT	517	VG	06/26/79		
HUBVIBVG	508	VG	04/10/79		07/19/79
OLSMPLOT	611	VG	07/30/79		03/26/80
OLSMANUP	707	VG	07/31/79		10/11/79
OLSTRIML	4430	VG	08/09/79	09/07/79	03/18/80
S32767	1208	DICK	03/06/80		04/23/80
C81COMPR	5631	DICK	04/01/80		06/22/80
AZIMTEST	30	DICK	04/23/80		04/23/80
BENDING	1335	DICK	06/02/80		06/23/80
XV15DATA	2064	BILGER	06/03/80		06/21/80
NOISE214	1357	DICK	06/05/80		06/23/80
FCPTEST	3711	DICK	12/08/79		04/11/80
ACOUST2	3159	J.T.BRIEGER	08/23/79		03/29/80
OLSMANUL	437	VG	08/27/79		10/11/79
ACOUSTIC	1111	J.T.BRIEGER	01/09/80	03/05/80	03/26/80
PARTNAME	1107	VG	02/22/80		04/19/80
ACCEPT	1485	VG	06/11/80		06/23/80

ENTER PARTITION NAME

Figure 31. Initialization phase menu of partitions.

EDIT specifies one of three Command Sequencing functions (NEW, CHANGE, or DELETE) as identified in the Input Substep. The meaning of these functions was covered in Section 4.6. Specific instructions for use of the CHANGE mode of EDIT to modify sequences of Command Steps are given in Section 5.6.1. The BUILD command was also covered in Section 4.6 along with SAVE, EXECUTE, and NOEDIT.

The COMMENT specification provides a user capability to enter additional labeling on plots and printouts beyond the labels normally provided by the program. The comment itself is entered as a special string in the Input Substep, which may contain blanks and commas but not slashes or single quotes. The comment is limited to 55 characters from the first nonblank character to the last nonblank character. The comment entry may not contain sequences of characters that create an invalid line error. For example:

COMM/THIS IS THE 2ND FILTERED CASE/

would be unacceptable because "2ND" creates an invalid line error. However, this comment could be successfully entered by enclosing it in single quotes. Thus:

COMM/'THIS IS THE 2ND FILTERED CASE'/

is acceptable. The comment may appear in a line separate from the Specification Substep. Thus:

COMMENT/

THIS IS ONE VERY LONG COMMENT USING FIFTY-FIVE COLUMNS/

is acceptable.

The MENU specification allows the user to request displays helpful in determining how to process the data. One such MENU is a list of the Groups present on the Info File. All the possible MENU's are listed with explanation in Section 5.4.

The SET specification is provided to change certain run settings. The settings that can be changed include: Rotor Mode, Screen Plot Size, Plot Grid Mode, Plot Tics Mode, Plot Label Speed, and Plot Copy Mode. The necessary keyword to specify a setting change is entered in the Action Substep and the possible entries are listed in Section 5.3.

The UTILITY specification is provided for several unrelated actions by the program. The possible actions are to mask or unmask an item code, to change the Master File partition that

is being accessed or to access an additional partition, or to save a plot that has been drawn in the interactive or interactive graphics mode. The possible utility actions are discussed in Section 5.3.

The TERMINATE specification causes the program to halt. A message is printed:

RUN TERMINATES

and the program is exited.

5.3 ACTION SUBSTEP COMMANDS

The Action Substep entries define the process requested by an ANALYZE, DERIVE, SET, or UTILITY Specification Substep. Many of the initial Action Substep entries are followed by additional qualifying entries. Some of these qualifiers have default values. Each initial Action Substep entry is explained here along with qualifiers and defaults for qualifiers. Parentheses in the listing only signify that a number or string should be substituted in the corresponding entry location. The parentheses do not appear in the actual command step.

5.3.1 ANALYZE Commands

Table 6 lists all of the ANALYZE Action Substep entry options. In this table, an open bracket indicates a branch option. Following is a description of the specific meaning for each entry.

HARMONIC, (Number of Harmonics), (Harmonic Number)

This command generates a Harmonic Analysis for an integral number of rotor cycles. The number of cycles is defined in the Input Substep. The second entry in the Action Substep gives the number of Harmonics to extract. The zeroth harmonic (mean level) is not counted in this number, but it is always calculated unless the number of harmonics is one. The third entry in this substep (Harmonic Number) gives the specific Harmonic number to calculate if the second entry is one. Harmonic analysis produces two double-row elements (TOP = amplitude, BOTTOM = phase) for output. The rotor cycles can be defined by the main or tail rotor as selected at the beginning of the program run (see Section 5.1) or by the SET command.

TABLE 6. 'ANALYZE' ACTION SUBSTEP ENTRY OPTIONS

- ANALYZE/
  - HARMONIC (# HARMONICS) (HARMONIC NUMBER) /...
  - FILTER (UPPER BREAK FREQ) (LOWER BREAK FREQ) (#POLES) /...
  - SPECTRUM (MAX FREQ) (WINDOW) /...
  - DAMPING (DAMPING FREQ) /...
  - AVERAGE /...
  - MMAX /...
  - RESPONSE (MAX FREQ) (WINDOW) (ADJ POINT AVERAGE) /...
  - COHERENCE (MAX FREQ) (WINDOW) (ADJ POINT AVERAGE) /...
  - AUTO
    - DENSITY (MAX FREQ) (WINDOW) (ADJ POINT AVERAGE) /...
    - CORRELATION (MAX OFFSET) (RECTIFICATION) (NORMALIZATION) /...
  - CROSS
    - DENSITY (MAX FREQ) (WINDOW) (ADJ POINT AVERAGE) /...
    - CORRELATION (MAX OFFSET) (RECTIFICATION) (NORMALIZATION) /...
  - STATISTIC
    - MEAN /...
    - VARIANCE /...
    - DEVIATION /...
    - FIT (NUMBER OF BINS) /...
  - ACOUSTIC
    - NARROW (FILTER BANDWIDTH) (CORRECTION LEVEL) /...
    - OCTAVE (CORRECTION LEVEL) /...
    - THIRD (CORRECTION LEVEL) /...
    - PNL (CORRECTION LEVEL) /...
    - DBA (CORRECTION LEVEL) /...
    - DBB (CORRECTION LEVEL) /...
    - DBC (CORRECTION LEVEL) /...
    - DBD (CORRECTION LEVEL) /...

The default for the third entry is one and the default for the second entry is twelve. No more than 100 Harmonics may be specified. Examples of use of this substep are:

ANALYZE/HARMONIC 6/...

or

ANAL/HARM, 1, 4/...

FILTER, (Upper Break Freq), (Lower Break Freq), (Number Poles)
This command causes input time histories to be digitally filtered. The pass band is the frequency interval between the frequencies: (Upper Break Freq) and (Lower Break Freq) in Hz. If (Lower Break Freq) is set to zero, the filter becomes a low pass filter. The rate of roll-off (see Paragraph 6.1.2) outside the pass band is set by (Number Poles). Increasing the number of poles in the filter increases the roll-off rate. Normally three or four poles are adequate. The allowed range for (Upper Break Freq) is between 0.1 and 100,000 Hz with no default. The allowed range for (Lower Break Freq) is between 0.0 Hz and (Upper Break Freq) with a default of 0.0 Hz. The allowed (Number Poles) is two through seven with a default of four. Examples of this substep command are:

ANALYZE/FILTER, 50, 0, 3/...

ANAL/FILT 100/...

SPECTRUM, (Maximum Frequency), (Window)
This command generates an Amplitude Spectrum of a measured or processed time history. Frequency components are displayed between the reciprocal of half the record length and (Maximum Frequency) in Hz. (Maximum Frequency) must be between 0.0 and 100,000 Hz. If (Maximum Frequency) is set to 0.0 or is greater than the Nyquist Frequency for data, the corresponding limit is set to the Nyquist Frequency during the processing phase. (Window) specifies a data window function to be applied to the time history before the spectrum is calculated (see Section 6.1). The allowed windows are COSINE TAPER, HANNING and NONE. The default (Maximum Frequency) is 0.0, which is reset to the Nyquist Frequency in processing. The default for (window) is COSI (cosine taper). Examples:

```
ANALYZE/SPECTRUM 200 NONE/...
```

```
ANALYZE/SPEC/...
```

DAMPING, (Damping Frequency)

This process calculates an estimate of the percentage of critical damping found in a time history for the specified frequency (Damping Frequency). Damping analysis produces a single output value - percentage of critical damping - for each individual time history that is input. Thus, a DAMPING analysis of a single time history cannot be plotted (the value can be printed). DAMPING analysis must be performed on a group of time histories to generate sufficient points for a plot. The range of (Damping Frequency) is 0.0 through 100,000, but a frequency higher than the Nyquist Frequency for the data will produce a processing error (number 771). There is no default for (Damping Frequency). An example:

```
ANAL/DAMPING 112/...
```

AVERAGE

This process creates one representative time history, one rotor cycle in length, by averaging together several contiguous rotor cycles of data from one sensor. When the process is performed, the sample rate for the output is arbitrarily set so that there are 256 output samples in the single cycle of data produced. Main or tail rotor azimuth can be used for specifying these data cycles as specified by the rotor mode selected at the beginning of the program run (see Section 5.1) or by use of the SET command. An example:

```
ANAL/AVERAGE/...
```

MMAX

This process generates min/max data from time history data. For every rotor cycle of each input time history, the minimum (min) and maximum (max) values are found. From these values, mean and oscillatory values are calculated (see Section 2.4 or 6.1). The oscillatory is output as the TOP double-row element and the mean as the BOTTOM. Rotor cycles are defined by the main or tail rotor as specified at the start of the program run (see Section 5.1) or by use of the SET command. The output is a time history with the first value located in time at the midpoint of the first cycle and having a data interval one rotor cycle in length. An example:

ANAL/MMAX/...

RESPONSE (Max Frequency) (Window) (Adjacent Point Average)
This process calculates the Frequency Response for one or more pairs of input and corresponding output functions of time for a linear system. Frequency components are displayed between the reciprocal of the record length and (Max Frequency) in Hz. (Max Frequency) must be between 0.0 and 100,000 Hz. If (Max Frequency) is set to 0.0 or is greater than the Nyquist Frequency for the input sample rate, then the corresponding limit is set just below the Nyquist Frequency during the processing phase. (Window) specifies a data window function to be applied to the time history before the Response is calculated (see Section 6.1). The allowed windows are COSINE TAPER, HALF COSINE, HANNING, AND NONE. (Adjacent Point Average) is a number that specifies the amount of adjacent frequency averaging that should be performed during processing. The number zero means no adjacent point averaging should be performed. A positive integer entry means that this number of adjacent frequency components will be taken in each direction to form an average frequency component. Thus, if two is entered, then five adjacent components will be averaged together to replace the center component of the five. The default (Max Frequency) is 0.0, which is reset to the Nyquist Frequency in processing. The default for (Window) is HALF (Half Cosine). The default for (Adjacent Point Average) is zero.
Examples:

ANALYZE/RESONSE 100 HANN 3/...

ANAL/RESP,,,5/...

COHERENCE (Max Frequency) (Window) (Adjacent Point Average)
This process estimates the Coherence Function from two or more pairs of functions of time. Coherence estimates are displayed between the reciprocal of the record length and (Max Frequency) in Hz. The entries (Max Frequency), (Window), and (Adjacent Point Average) are explained under the RESPONSE entry and the defaults are the same.
Examples:

ANALYZE/COHERENCE 50 COSI 2/...

ANALYZE/COHE 50/...

AUTO DENSITY (Max Frequency) (Window) (Adjacent Point Average)
This process estimates the Auto-Spectral Density or Power Spectrum from one or more functions of time.

Density estimates are displayed between the reciprocal of the record length and (Max Frequency) in Hz. The entries (Max Frequency), (Window), and (Adjacent Point Average) are explained under the RESPONSE entry and the defaults are the same. Examples:

ANALYZE/AUTO DENSITY 200 HALF 5/...

ANAL/AUTO DENS,,, 3/...

AUTO CORRELATION (Max Offset) (Normalization)
This process estimates the Auto-Covariance for one or more functions of time. Correlation estimates are displayed for offsets between zero and (Max Offset) in seconds. (Max Offset) must be between 0.0 and 1000.0 seconds. If (Max Offset) is set to 0.0, then this limit is reset to half the record length during the processing phase. If (Max Offset) is greater than the record length, then this limit is reset to the record length. Two keyword entries are allowed for (Normalization), NORMALIZE or NONORMALIZE. For NONORMALIZE, the ordinary Auto-Covariance is calculated. For NORMALIZE, the Auto-Covariance is normalized by dividing the value at every offset by the value at the zero offset. The default for (Max Offset) is 0.0 and for (Normalization) the default is NORMALIZE. Examples:

ANALYZE/AUTO CORRELATION .2 NONORM/...

ANAL/AUTO CORR,, NONORM/...

CROSS DENSITY (Max Frequency) (Window) (Adjacent Point Average)
This process estimates the Cross-Spectral Density or Cross-Power Spectrum from one or more pairs of functions of time. Density estimates are displayed between the reciprocal of the record length and (Max Frequency) in Hz. The entries (Max Frequency), (Window), and (Adjacent Point Average) are explained under the RESPONSE entry and the defaults are the same. Examples:

ANALYZE/CROSS DENSITY 100 NONE 2/...

ANAL/CROSS DENS/...

CROSS CORRELATION (Max Offset) (Normalization)
This process estimates the Cross-Covariance for one or more pairs of functions of time. Correlation estimates are displayed for offsets between minus (Max Offset)

and plus (Max Offset) in seconds. The entry (Max Offset) is explained under the AUTO CORRELATION entry and the default is the same. (Normalization) is similar to normalization for Auto-Correlation but users should consult Section 6.1 for specific interpretation of this entry. NORMALIZE is the default for the (Normalization) entry. Examples:

ANALYZE/CROSS CORRELATION .4 NONORMALIZE/...

ANAL/CROSS CORR ,, NONORM/...

STATISTIC MEAN
This process calculates the mean for one or more functions of time. Example:

ANALYZE/STAT MEAN/...

STATISTIC VARIANCE
This process estimates the variance from one or more sample functions of time. Example:

ANALYZE/STAT VAR/...

STATISTIC DEVIATION
This process estimates the standard deviation from one or more sample functions of time. Example:

ANAL/STATISTIC DEV/...

STATISTIC FIT (Number of Bins)
This process tests the goodness of fit to a normal distribution of the sample values in one or more sample functions of time. The Chi-squared test is used. Consult Section 6.1 for specific interpretation of the output value. (Number of Bins) is one more than the number of degrees of freedom for the test. (Number of Bins) may be a number between and including 3 and 100. However, an error is returned if too many bins are specified for the number of input values (see Section 6.1). Default for (Number of Bins) is seven. Examples:

ANALYZE/STATISTIC FIT 11/...

ANAL/STAT FIT/...

ACOUSTIC NARROW (Filter Bandwidth) (Correction Level)
This process is Narrow Band Analysis of acoustic data. Output is a function of frequency with a domain between slightly more than half the (Filter Bandwidth) and slightly less than the Nyquist Frequency for the input

sample rate. (Filter Bandwidth) is the pass band width in Hz for the constant width filter used to scan the data. Possible values for this entry are 1 to 99 but the bandwidth must not be less than twice the reciprocal of the input record length. The (Correction Level) is a sound level in dB that is added to each value of the resultant function of frequency. Allowed values for (Correction Level) are between -100 and 100. The default (Filter Bandwidth) is 8.0 and the default (Correction Level) is 0.0. Examples:

ANALYZE/ACOUSTIC NARROW 16 2.3/...

ANAL/ACOU NARR/...

ACOUSTIC OCTAVE (Correction Level)
This process is Octave Analysis of acoustic data. (Correction Level) is explained under the ACOUSTIC NARROW entry with the same default. Examples:

ANAL/ACOUSTIC OCTAVE 1.4/...

ANAL/ACOU OCT/...

ACOUSTIC THIRD (Correction Level)
This process is Third Octave Analysis of acoustic data. (Correction Level) is explained under the ACOUSTIC NARROW entry with the same default. Examples:

ANALYZE/ACOUSTIC THIRD .8/...

ANAL/ACOU THIRD/...

ACOUSTIC PNL (Correction Level)
This process is Perceived Noise Level calculation for acoustic data. (Correction Level) is explained under the ACOUSTIC NARROW entry except that this level is added to the Third Octave levels before the final derivation of PNdB. Default for (Correction Level) is zero. Examples:

ANALYZE/ACOUSTIC PNL .6/...

ANAL/ACOU PNL/...

ACOUSTIC [DBA, DBB, DBC, OR DBD] (Correction Level)
This process is one of the weighted integrations of noise level using the "A", "B", "C", or "D" weighting networks. (Correction Level) is explained under the ACOUSTIC NARROW entry and the default is the same. Examples:

ANALYZE/ACOUSTIC DBA .8/...

ANAL/ACOU DBC .4/...

ANAL/ACOU DBD/...

5.3.2 DERIVE Commands

Table 7 lists all of the DERIVE Action Substep entry options. In this table, an open bracket indicates a branch option. Following is a description of the specific meaning for each entry.

TAS, (OAT), (Static Pressure), (Slope), (Intercept)

This process derives Vehicle True Airspeed from measured indicated airspeed and the parameters specified by the qualifiers above according to the method described in Section 6.2. Alternatively, for True Airspeed provided directly in Knots (e.g., from a simulation program), the input values are simply read and smoothed. For this case, the (OAT), (Static Pressure), (Slope), and (Intercept) are ignored.

For input of measured indicated airspeed in knots squared, a derivation must be performed and the above entries are used. Entry number two, Outside Air Temperature (OAT in degrees Celsius) and entry number three, Static Pressure (PSIA), can be specified either with a constant numerical value or the word CALCULATE, which means that smoothed, measured data will be used for these parameters (see Section 2.8). Both of these entries default to 'CALC' initially and then to the previous setting. Static Pressure and OAT are common to many derived parameter calculations, and the previous setting default means the setting specified in the most recent such derivation. Entries four and five correspond to slope and intercept values for the conversion from True Indicated Airspeed to Calibrated Airspeed (see Section 6.2). The initial defaults for these numbers are 1.0 and 0.0, respectively. When new values are specified for these two entries, these new values become the defaults until reset or until the end of the program run. Allowed values for the slope are .5 to 2.0 and allowed intercept values are -40.0 to +40.0.

TAS is an 'attached parameter' (see Section 2.8) and has the following attached parameter characteristics: (1) once calculated for a given counter and time span, the parameter is not recalculated until a new counter or time span is specified, (2) the parameter values are smoothed

TABLE 7. 'DERIVE' ACTION SUBSTEP ENTRY OPTIONS

```
        TAS (OAT) (STATIC PRESSURE) (SLOPE) (INTERCEPT) /...
        MRPM /...
        MSHP /...
        CP (ROTOR RADIUS) (OAT) (STATIC PRESSURE) /...
        CN /...
        CC /...
        CM /...
DERIVE  MCT (OAT) (STATIC PRESSURE) (WEIGHT OR FORCE) (ROTOR RADIUS) /...
        MCQ (OAT) (STATIC PRESSURE) (ROTOR RADIUS) /...
        MFLO (OAT) (STATIC PRESSURE) (ANGLE) /...
        DFLO (OAT) (STATIC PRESSURE) ANGLE) /...
        BLDISP (HARMONIC NUMBER) /...
        SLOPE (ROTOR RADIUS) /...
        DENALT (OAT) /...
        MRAZ /...
```

a great deal, and (3) the parameter output data stream contains one value for each rotor cycle.

Here are some examples of TAS commands:

DERIVE/TAS CALC 14.23/...

DERIVE/TAS CALC CALC 1.014 -.013/...

DERI/TAS/...

MRPM

This process calculates the rotor speed in units of RPM. The main or tail rotor data will be displayed based on the 'rotor mode' defined at the beginning of the program run (see Section 5.1) or by use of the SET command. An example:

DERI/MRPM/...

MSHP

This process calculates the Mast Horsepower as described in Sections 2.5 and 6.2. The shaft torque item code as well as the rotor azimuth item code must be present for this calculation. The main or tail rotor will be selected as above. Example:

DERI/MSHP/...

CP, (Rotor Radius), (OAT), (Static Pressure)

This process calculates Blade Static Pressure Coefficient from Blade Absolute Pressure Sensor data along with all of the attached parameters (Section 2.8), rotor radius, and the radial station on the blade expressed as a fraction of the total radius. See Section 6.2 for a description of the process. The second entry is rotor radius in inches from hub to tip. There is no initial default for this value but, once set, the most recently entered value becomes the default. The most recent rotor radius entry in any of several derivations will be used as the default.

The third and fourth entries, OAT and Static Pressure, are the same as the OAT and Static Pressure entries for the True Air-speed (TAS) derivations. Examples:

DERI/CP, 264, CALC, 14.23/...

DERI/CP,,,14.50/...

DERI/CP/...

CN

The Blade Normal Force Coefficient is calculated from Blade Static Pressure Coefficient, C_p, data integrated around all the chord positions for a radial station. The only possible input source for the C_n process is a scratch file (SCF1, SCF2 or SCF3) containing C_p data. The command step that generated these C_p data must have used the Info File to generate values from a sufficient number of sensor positions around the chord for a C_n integration to be successful. Example:

DERI/CN/...

CC

The Blade Chordwise Force Coefficient is calculated from C_p data in similar fashion to C_N. Example:

DERI/CC/...

CM

The Quarter Chord Blade Pitching Moment Coefficient is calculated from C_p data in similar fashion to C_N. Example:

DERI/CM/...

MCT,(OAT),(Static Pressure),(Weight or Force),(Rotor Radius)

This process calculates the Thrust Coefficient as described in Section 6.2. Entries two and three, OAT and Static Pressure, are as described under the True Airspeed (TAS) derivation. The fourth entry is Gross Vehicle Weight (GVW) or antitorque force in pounds. There is no initial default for (Weight or Force) but, once set, the most recently entered value becomes the default. The fifth entry, Rotor Radius, is described under the C_p derivation. Examples:

DERI/MCT 30 14.4 8300 264/...

DERI/MCT,,, 9000/...

DERI/MCT/...

MCQ, (OAT), (Static Pressure), (Rotor Radius)

The Torque Coefficient, C_Q, is calculated as described in Section 6.2. The second and third entries, OAT and Static Pressure, are described under the TAS derivation. The fourth entry, Rotor Radius, is described under the C_p derivation. Examples:

DERI/MCQ, CALC, CALC, 264/...

DERI/ MCQ, 32.5,, 264/...

DERI/MCQ/...

MFLO, (OAT), (Static Pressure), (Angle)

The Blade Local Flow Magnitude (in ft/sec) is calculated as described in Section 6.2. When this command is selected, both the Flow Magnitude and Direction (in degrees) are calculated with the Magnitude output as the TOP double-row element and the Direction as the BOTTOM double-row element. When the output is printed, both double-row elements are shown. Both double-row elements may be plotted by using the DPLOT disposition option.

Entries two and three, OAT and Static Pressure, are described under the TAS derivation. The fourth entry, (Angle), is the angle in degrees formed by the inboard pointing Boundary Layer Button (BLB) sensor and the chordline. The initial default for (Angle) is 45 degrees. This number is replaced by any nondefault entry. Examples:

DERI/MFLO, CALC, 14.3, 43.5/...

DERI/MFLO/...

DFLO, (OAT), (Static Pressure), (Angle)

The Blade Local Flow Direction command is identical to the 'MFLO' command except that the Flow Direction becomes the TOP output double-row element for plotting output and the Flow Magnitude becomes the BOTTOM. The DPLOT option should not be used for this derivation. Instead, use the MFLO derivation. On output, positive angles mean outboard to inboard flow and negative angles mean inboard to outboard. Angles are measured in degrees from the chordline. Examples:

DERI/DFLO CALC CALC 46.1/

BLDISP, (Harmonic Number)

Local Blade Displacement, in inches, is calculated from accelerometers on the blade, which are oriented in either the chordwise or beamwise direction. The displacement is calculated for a single harmonic of the rotor cycle according to the process described in Section 6.2. Section 6.2 also documents deficiencies in this derivation method. The second entry, (Harmonic Number), is the single rotor cycle harmonic number and has allowed values

of 1 through 24. The default harmonic number is 1. One complete rotor cycle of 256 displacement values is always produced by this derivation. Examples:

DERI/BLDIS 2/...

DERI/BLDIS/...

SLOPE, (Rotor Radius)

Local Blade Slope is calculated from Local Blade Displacement derivations for multiple radial positions on the blade. Thus, the input source for this process must be a scratch file (SCF1, SCF2 or SCF3). The output of the process is unitless (inches/inch). The second entry, (rotor radius), is as described under the C_p derivation. Examples:

DERI/SLOPE 264/...

DERI/SLOPE/...

DENALT, (OAT)

Density Altitude is calculated from Boom System Static Pressure and OAT according to the process described in Section 6.2. The second entry, (OAT), is as described under the TAS derivation. Example:

DERI/DENALT 29.5/...

DERI/DEN/...

MRAZ

Rotor Azimuth is derived from the rotor azimuth item code as described in Section 6.2. This parameter is displayed as a ramp function ranging from 0.0 to 360 degrees. Either the main or tail rotor azimuth will be displayed according to the 'rotor mode' specified at the beginning of the program run (see Section 5.1) or by use of the SET command. Example:

DERI/MRAZ/...

5.3.3 SET and Utility Commands

Table 8 lists all of the SET and UTILITY Action Substep entry options. Input Substep entry options for these commands are shown as well since the command structures are uncomplicated. Open brackets indicate branch options in this table. Following is a description of the specific meaning for each entry.

TABLE 8. "SET' AND 'UTILITY' ACTION SUBSTEP ENTRIES

SET/	MAIN/	
	TAIL/	
	GRID/	
	NOGRID/	
	TICS/	
	NOTICS/	
	HALF/	
	FULL/	
	QUICK/	
	SLOW/	
	COPY/	
	NOCOPY/	
UTILITY/	PARTITION/(PARTITION NAME)	FIRST/
		SECOND/
	MASK/(ITEM CODE)/	
	UNMASK/(ITEM CODE)/	
	COPY/	

SET/[MAIN or TAIL]/

This command sets the Rotor Mode as discussed in Section 5.1. MAIN selects Main Rotor Azimuth to be used for input definition, processing, and display functions. TAIL selects Tail Rotor Azimuth for these functions. Example:

SET/TAIL/

SET/[GRID or NOGRID]/

This command sets the Plot Grid Mode as discussed in Section 5.1. GRID selects a gridded plotting area for X-Y plots. NOGRID selects X-Y plots without grid reference lines. Example:

SET/NOGRID/

SET/[TICS or NOTICS]/

This command sets the Plot Tics Mode as discussed in Section 5.1. TICS selects an X-Y plotting mode with extensive tic marks around the boundary to define the scale. NOTICS selects a mode without these marks. If both NOTICS and NOGRID are selected, then tic marks will still be drawn for annotated scale intervals. Example:

SET/TICS/

SET/[HALF or FULL]/

This command sets the Tektronix Plot Size Mode. HALF is the normal screen apportionment as displayed previously in Figure 27. For the FULL setting, the plotting area (i.e., the area that actually contains the curves) is expanded to occupy as much of the Tektronix screen area as possible while retaining the same proportion for the plot. The FULL setting is incompatible with the QUICK setting and/or the COPY setting so that if FULL is set, a QUICK setting is automatically reset to SLOW and the COPY setting is automatically reset to NOCOPY. This command is valid only for the Interactive Graphics mode of operation. Example:

SET/FULL/

SET/[QUICK or SLOW]/

This setting is only valid for certain installations of DATAMAP and may be used only in the Interactive Graphics mode of operation. With the SLOW setting, character annotation that is drawn in plots is generated by individual strokes. With the FAST setting, this character annotation is generated with alphanumeric writes so that

labeling is drawn much more quickly. Non-character output to plots is not affected by this setting. The QUICK setting is incompatible with the FULL setting and/or the COPY setting so that if QUICK is set then a FULL setting is automatically reset to HALF and a COPY setting is automatically reset to NOCOPY. Example:

SET/QUICK/

SET/[COPY or NOCOPY]/
This setting is valid only for certain installations of DATAMAP. Copy may be set for either the Interactive or Interactive Graphics mode of operation, but may not be set for the Batch mode of operation. When COPY is set, a copy of each plot frame subsequently generated is saved temporarily on a disc file. Each new frame overwrites the previous frame. For a frame to be permanently saved, the UTILITY COPY command must be used to transfer the temporary frame to a disc file that can accommodate more than one plot frame. Plotting speed may be degraded when the COPY mode is set. The COPY setting is incompatible with the FULL setting and/or the QUICK setting so that if COPY is set, then a FULL setting is automatically reset to HALF and a QUICK setting is automatically reset to SLOW. Consult local documentation to determine whether COPY may be set for the particular installation of DATAMAP. Example:

SET/COPY/

UTILITY/PARTITION/(Partition Name) (FIRST or SECOND)/
This command will change a partition that is accessed or provide access to a second partition. The partition that is specified for access in the program run Initialization Phase (see Section 5.1) is considered to be the FIRST accessed partition. The Processing Program can access two partitions simultaneously and these "access slots" are labeled the FIRST and the SECOND. When data are required, as specified by item code and counter, the FIRST partition is searched for the data. If the data are not found on the FIRST partition, the SECOND partition is searched. Thus, if an item code/counter pair occurs on both partitions, the data from the FIRST partition will be used. The (FIRST or SECOND) entry specifies which access slot will be changed or filled. It is acceptable for either or both access slots to be filled. The (Partition Name) entry specifies which partition will either fill the empty slot or replace the partition that occupies that slot. NONE is an acceptable entry for (Partition Name) and specifies that access to the partition that occupies the indicated access slot should be severed. Examples:

```
UTILITY/PARTITION/JONES5 SECOND/

UTIL/PART/NONE SECO/
```

UTILITY/MASK/(Item Code)/
This command will mask the item code named by the (Item Code) entry. The effect of the mask is that the item code will not be found for subsequent searches of the Master File during the current program run, unless the UNMASK command is used for the item code. This command is useful when groups of item codes are referenced and the data for one or more of these item codes are invalid. As many as 64 item codes may be masked at one time, but a separate MASK command is required for each item. Masking an item code in one run of the Processing Program has no effect on other Processing Program runs occurring simultaneously. Example:

```
UTIL/MASK/R849/
```

UTILITY/UNMASK/(Item Code)/
This command removes the effect of a previous MASK command on the named (Item Code). Example:

```
UTIL/UNMASK/R849/
```

UTILITY/COPY/
This command is valid only for certain installations of DATAMAP and only for the Interactive or Interactive Graphics mode of operation. When this command is entered, the plot frame that is currently stored on the temporary single plot frame storage file is transferred to the multiple frame storage file for later plotting on a batch graphics device. Consult the SET/COPY/ command for additional descriptions of these files. For a plot frame to be copied, a plot must be generated while the copy mode is set. Following is an example of a sequence of commands that uses the COPY capability.

```
SET/COPY/

DISP/......./PLOT.../

UTIL/COPY/

DISP/....../MPLOT.../

UTIL/COPY/
```

5.4 INPUT SUBSTEP COMMANDS

The sequence of entries required for the Input Substep depends on the preceding Specification and Action Substep entries. In addition, this sequence can branch, depending upon the Input Substep entries themselves. For example, the first Input Substep entry after the sequence

DERI/MRPM/...

is the counter for the input data stream, while for the sequence

ANAL/MMAX/

the first Input Substep entry is a source of data. This source could be an individual item code, a group of items specified in the Info File, or a scratch file (SCF1, SCF2 or SCF3). For each of those possibilities, there is a different sequence of subsequent entries.

The Input Substep sequences are displayed in Table 9 using short descriptors for the meaning of each entry position. The meaning of every entry descriptor used in the table and possible inputs for each are listed below. Specification and Action Substep combinations that lead to like Input Substep entry sequences are grouped together in the table with a closed bracket joining the paths in the Input Substep. Open brackets indicate a branch in paths depending upon a particular entry. For each open bracket in the Input Substep, a description is provided over the column of possible entries. This descriptor is enclosed in double asterisks (e.g., **SOURCE**). The possible entries in the column are described in the corresponding descriptor definition.

(Angle) - Rotor azimuth in degrees for single azimuth position input. The start point and length of measured data input time histories is sometimes defined as an integer number of complete rotor cycles. If zero cycles are selected, then a single azimuth position defined by (Angle) is input. The default (Angle) is -.01 degree. If a nonzero number of cycles is selected, then (Angle) has no meaning and should be defaulted.

AVERAGE - Ensemble averaging selection. Ensemble averaging is selected with the keyword "ENSEMBLE" and individual data stream processing is selected with the keyword "INDIVIDUAL". For the "ENSEMBLE" selection, input must be from a

TABLE 9. INPUT SUBSTEP SEQUENCES

```
ANAL/HARMONIC.../  )
ANAL/AVERAGE.../    }
ANAL/MMAX/          }
DERI/BLDIS.../     )

  **SOURCE**
 ( (ITEM CODE) (COUNTER) (TIME) (CYCLES) (ANGLE)/...
 ( (GROUP) (NAME) (DBLROW) (3RD DIM) (2ND DIM) (COUNTER) (TIME)

 ( SCF1 ) **1ST DIM**
 ( SCF2 }  { (VALUE) (1ST VAR) (2ND DIM) (3RD DIM) (DBLROW)/...
 ( SCF3 )  { ALL (2ND DIM) (3RD DIM) (DBLROW)/...

  (CYCLES) (ANGLE)/...


DISPLAY/             )
ANAL/FILT.../         }
ANAL/SPECT.../        }
ANAL/DAMP.../         }
ANAL/ACOUSTIC.../    )

  **SOURCE**
 ( (ITEM CODE) (COUNTER) (TIME) (DURATN)/...
 ( GROUP (NAME) (DBLROW) (3RD DIM) (2ND DIM) (COUNTER) (TIME) (DURATN)/...

 ( SCF1 ) **1ST DIM**
 ( SCF2 }  { (VALUE) (1ST VAR) (2ND DIM) (3RD DIM) (DBLROW)/...
 ( SCF2 )  { ALL (2ND DIM) (3RD DIM) (DBLROW)/...


                   **IN 1**  **IN 2**
                   { SCF1 }  { SCF2 }
ANAL/COHERENCE.../ { SCF2 }  { SCF2 } (2ND DIM) (DBLROW)/...
                   { SCF3 }  { SCF3 }


                                    **IN 1**  **IN 2**
                     **AVERAGE**    { SCF1 }  { SCF1 }
                    ( ENSEMBLE      { SCF2 }  { SCF2 } (2ND DIM) (DBLROW)/...
      { CROSS.../ }  (               { SCF3 }  { SCF3 }
ANAL/ {           }  (                **IN 1**  **IN 2**
      { RESP..../ }  (               { SCF1 }  { SCF1 }
                    ( INDIVIDUAL    { SCF2 }  { SCF2 } (2ND DIM) (3RD DIM) (DBLROW)/...
                                    { SCF3 }  { SCF3 }
```

TABLE 9. INPUT SUBSTEP SEQUENCES (Continued)

```
ANAL/RESP.../  }
ANAL/CROSS.../ }

  **AVERAGE**   **SOURCE**
  ENSEMBLE      { SCF1 }
                { SCF2 } (2ND DIM) (DBLROW)/...
                { SCF3 }

                **SOURCE**
                ( (ITEM CODE) (COUNTER) (TIME) (DURATN)/...
  INDIVIDUAL    { GROUP (NAME) (DBLROW) (3RD DIM) (2ND DIM) (COUNTER)
                { SCF1 }
                { SCF2 } (2ND DIM) (3RD DIM) (DBLROW)/...
                ( SCF3 }

     (TIME) (DURATN)/...


        ( TAS.../    )
        ( MRPM/      )
        ( MSHP/      )
DERI/   { MRAZ/      } (COUNTER) (TIME) (CYCLES) (ANGLE)/...
        ( MCQ.../    )
        ( MCT.../    )
        ( DENALT.../ )


DERI/MFLO.../ }
DERI/DFLO.../ }

  **(SOURCE)**
  ( GROUP (NAME) (3RD DIM) (2ND DIM) (COUNTER) (TIME) (CYCLES) (ANGLE)/...
  {
  { SCF1   **(1ST DIM)**
  { SCF2   { (VALUE) (1ST VAR) (2ND DIM) (3RD DIM)/...
  ( SCF3   { ALL (2ND DIM) (3RD DIM)/...


              **SOURCE**
              ( (GROUP) (NAME) (DBLROW) (3RD DIM) (2ND DIM) (COUNTER) (TIME)
              {
DERI/CP.../   { SCF1 }   **1ST DIM**
              { SCF2 }   { (VALUE) (1ST VAR) (2ND DIM) (3RD DIM) (DBLROW)/...
              ( SCF3 }   { ALL (2ND DIM) (3RD DIM) (DBLROW)/...

     (CYCLES) (ANGLE)/...
```

TABLE 9. INPUT SUBSTEP SEQUENCES (Concluded)

```
DERI/SLOPE.../    **SOURCE**
DERI/CN/           SCF1   **1ST DIM**
                   SCF2    (VALUE) (1ST VAR) (3RD DIM)/...
DERI/CC/           SCF3    ALL (3RD DIM)/...
DERI/CM/


      **EDIT FUNCTION**
       NEW
EDIT/  CHANGE  (BLOCK NAME)/
       DELETE


BUILD/(BLOCK NAME)/


EXECUTE/(BLOCK NAME) %(PARM 1) %(PARM 2) %(PARM 3) %(PARM 4)/


      **MENU TYPE**
       DATA/
       (COUNTER)/
       PARTITION/
MENU/  SCRATCH/
       INFO/
       MASK/
       EDIT/
       SET/


COMMENT/(COMMENT STRING)/
```

scratch file and the results from each column element are averaged together. Averaging is performed at the stage of each process appropriate for the analysis that is being performed (see Section 6.1). When "ENSEMBLE" is selected, no option is provided to select between ALL or one column element from the scratch file. When input is from two scratch files (e.g., for Cross-Spectral Density), then the 'ENSEMBLE' or 'INDIVIDUAL' selection applies to both scratch files.

(Block Name) - Four-character name of a command sequence. Each command sequence block generated under the command sequencing capability is named with a unique four-character string. Command sequences are referenced by name for creation, modification, execution, or deletion. There is no default for this entry.

(Comment String) - Sequence of up to 55 characters to be used for additional labeling of plots and printouts. The sequence may include blanks or commas but not slashes. Blanks or slashes may not isolate a string that forms a numeric error (e.g., THE 3RD FLIGHT). However, single quotes may be used around the whole comment to allow entries that would otherwise form line errors. When quotes are not used, the comment begins with the first nonblank character after the slash ending the specification substep and ends with the last nonblank character before the slash ending the input substep. The comment string may not extend from the end of one line to the beginning of the next but may occupy a full input line after the specification substep was entered on the previous line. The comment is initially blank. Once entered, a comment remains unchanged until a new comment step is executed. The Comment can be set to blanks with an Input substep containing only blanks.

(Counter) - Specifies the unique integer assigned to reference the data from some specific period of time when useful data were taken or from a simulated flight condition or maneuver. Allowed counter values are 1 through 999999. There is no initial default Counter, but, once set, the last Counter specified is the default.

(Cycles) - Integer number of rotor cycles of data input. For certain Specification and Action Substep combinations, the period of the input data record is defined as an integer number of contiguous rotor cycles. The number of cycles may be any positive integer up to 1000 or zero as indicated under (Angle). In selecting the number of cycles to process, the user must consider the limited amount of processing storage area in the Processing Program. The default number of cycles is one.

(Dbl Row) - Double-row option selection. When there are two available double-row elements for input, the user must specify whether to use both or a single element by entering 'BOTH', 'TOP, or 'BOTTOM.' The meaning of the latter two possible entries must be interpreted with regard to the type of data being input. The obvious connection of the words 'TOP' and 'BOTTOM' is to the upper and lower surfaces of the rotor blade. However, these entries may also apply, for example, to the amplitude and phase parts of stored Harmonic Analysis output. If 'BOTH' is entered, then both double-row elements will be input if available. If only one element is available, then only that element will be input. If 'TOP' or 'BOTTOM' is entered, then the corresponding double-row element must be present for input. 'BOTH' is the default for this entry.

(Duratn) - Length of input record in seconds. For certain Specification and Action Substep combinations, the period of the input data is defined as a length of time. This entry may range in value from 0.0 to 1000 seconds. If '0.0' is entered, then a single data point corresponding to (Time) will be input. When entering (Duratn), the user must consider the available data stored on the Master File and the storage limitations of the program. There is no default for this entry.

(Edit Function) - Command sequence process selected. This entry selects a specific command sequencing function after an 'EDIT' Specification Substep. There are three possible entries. If 'NEW' is specified, a new command sequence block is originated and all subsequent commands are

copied to that block and not executed until a 'NOEDIT' command is entered. 'CHANGE' specifies that a special processing mode will be entered where existing command sequence blocks can be modified. Special instructions for entering commands in this processing mode can be found in Section 5.7.1. 'DELETE' specifies that the listed command sequence block will be removed from storage and the block name and storage are freed for future use. There is no default for this entry.

IN 1, **IN 2** - Scratch file input and input order selection. When two input functions of time are required for a process (e.g., for Frequency Response), then this input must come from two scratch files. Time histories from the scratch file selected for the **IN 1** entry will be considered the first input functions, and time histories from the scratch file selected for the **IN 2** entry will be considered the second input functions. For example, for Frequency Response, the data from the **IN 1** entry is the "input" data and the data from the **IN 2** entry is the "output" data from the system under consideration. During processing, column, row, and double-row elements from each scratch file are matched. For example, the time history from the **IN 1** scratch file for column two, row one, TOP double-row element will be paired with the corresponding time history element from the **IN 2** scratch file. Accordingly, the number of rows, columns, double-rows, and time history lengths must be the same for the two scratch files selected. Specification of column, row, or double-row elements in subsequent entries applies to both scratch files selected so that it is not possible, for example, to pair row two of SCF1 and row three of SCF3. Also, the scratch files selected for **IN 1** and **IN 2** must be different. Hence, SCF2 could not be specified for both the **IN 1** and the **IN 2** entry.

Menu Type - Kind of Menu to be displayed. The user may have several different kinds of Menu Listings displayed for assistance in generating commands. There are seven keywords available to invoke these Menu's. "DATA" requests a listing of counters present on the Master File

partition(s) that are currently accessed by the Processing Program. "INFO" requests a listing by name of groups present on the Info File along with the keyword pointers and corresponding item codes listed in the initial section of this file. "EDIT" requests a listing by name of the command sequence blocks currently on the command sequence file. "SCRATCH" requests a listing of the current contents of each scratch file. The "PARTITION" keyword causes a list of the partitions on the Master File to be generated. This list includes partition size in records, date of creation, date data were last included, and last date of use. In addition, the listing shows which partitions are currently accessed by the Processing Program. The "MASK" menu is a listing of the item codes that are currently masked. The "SET" keyword causes a listing of the current run settings to be generated including both the settings that can be changed with the "SET" command and the settings that must be specified during the program initialization phase. The user can also enter a counter to obtain a listing of item codes present on the Master File partition for that counter along with the corresponding record length, the offset from the start of data on the original storage medium to the start of data on the Master File, the breakpoint for any digital filtering applied to the data in storing it on the Master File, and the sample rate for the data as stored on the Master File. There is no default for this entry. Figures 32 through 38 show sample menus.

(Name) - Info File Group Name. This entry gives the four-character name of the Info File Group that should be used to specify item codes for data retrieval and geometric information and labels for processing and output. There is no default for this entry.

(PARM n) - Parameter entries for command sequence execution. The character "n" represents the parameter number passed, which can be the first, second, third, or fourth. Parameters are strings or numbers that are passed for insertion in the command sequence to be executed. If "i" is an integer between one and four, then

```
 NEW STEP  STEP CPU  0.01 SEC  TOTAL CPU   3.6 SEC
MENU/DATA/
          COUNTER LIST
   563     610     611     612     613     614     615     629     630
   631

 NEW STEP  STEP CPU  0.01 SEC  TOTAL CPU   3.7 SEC
```

Figure 32. Menu of counters in currently-accessed partitions.

```
 NEW STEP  STEP CPU  0.01 SEC  TOTAL CPU   3.6 SEC
MENU/630/
    ITEM CODE LIST FOR COUNTER    630
 ITEM  SEC FILT  OFF  RATE    ITEM  SEC FILT  OFF  RATE
 A019   1.0 100   0.5  512.   A020   1.0 100   0.5  512.
 A302   1.0 100   0.5  512.   A304   1.0 100   0.5  512.
 A306   1.0 100   0.5  512.   A307   1.0 100   0.5  512.
 A308   1.0 100   0.5  512.   A309   1.0 100   0.5  512.
 A311   1.0 100   0.5  512.   A315   1.0 100   0.5  512.
 A316   1.0 100   0.5  512.   A318   1.0 100   0.5  512.
 A319   1.0 100   0.5  512.   A320   1.0 100   0.5  512.
 A321   1.0 100   0.5  512.   A600   1.0 100   0.5  512.
 A601   1.0 100   0.5  512.   R992   1.0  -1   0.5  512.

 NEW STEP  STEP CPU  0.06 SEC  TOTAL CPU   3.6 SEC
```

Figure 33. Menu of item codes for one counter.

MENU/PARTITIONS/

LIST OF PARTITIONS IN THE MASTER FILE

CURR ACCESS	PARTITION	LENGTH	USER NAME	DATE CREATED	LAST CHANGED	LAST ACCESSED
	D800RED	3607	R-PHILBRICK	09/23/78		10/04/79
	D800WHIT	7097	R-PHILBRICK	09/23/78		11/02/79
	PUSHOVER	315	R-PHILBRICK	01/17/79	07/19/79	10/23/79
	C810LS35	2153	VANGAASBEEK	02/15/79		04/07/80
	C810LS45	428	VANGAASBEEK	02/15/79		02/22/79
	MANUDATA	115	VG	04/10/79	04/27/79	04/27/79
	HQDMTEST	491	WALTERS	06/19/79		06/20/79
	OLSBLACC	8720	VG	06/26/79		06/28/79
	OLSHVFLT	517	VG	06/26/79		
	HUBVIBUG	508	VG	04/10/79		07/19/79
	OLSMPLOT	611	VG	07/30/79		03/26/80
	OLSMANUP	707	VG	07/31/79		10/11/79
	OLSTRIML	4430	VG	08/09/79	09/07/79	03/18/80
	S32767	1208	DICK	03/06/80		04/17/80
	MANU2221	340	VG	03/07/80		03/26/80
	MANX2221	340	VG	03/08/80		03/26/80
	FCPTEST	3711	DICK	12/08/79		04/11/80
	ACOUST2	3159	J.T.BRIEGER	08/23/79		03/29/80
	OLSMANUL	437	VG	08/27/79		10/11/79
	ACOUSTIC	1111	J.T.BRIEGER	01/09/80	03/05/80	03/26/80
(2)	PARTNAME	1107	VG	02/22/80		04/18/80
(1)	C81COMPR	5631	DICK	04/01/80		04/18/80

NEW STEP STEP CPU 0.13 SEC TOTAL CPU 2.4 SEC

Figure 34. Sample menu of partitions.

10
F

```
SCF1

DERIVED PARAMETER:
NORMAL FORCE COEFFICIENT
DOUBLEROW - TOP              COUNTER -    611
1ST DIMENSION -     TIME (SECONDS)       256 POINTS  0.186 SECONDS
3RD DIMENSION (COLUMN POSITION) - FRACTN OF RADIUS
    0.400E+00  0.750E+00  0.864E+00  0.955E+00

SCF2

CYCLE AVERAGE:
RTR 1, BLD 1, NORMAL FORCE COEFFICIENT
DOUBLEROW - TOP              COUNTER - 360084
1ST DIMENSION -     TIME (SECONDS)       256 POINTS  0.185 SECONDS
3RD DIMENSION (COLUMN POSITION) - RADIUS IN FEET
    0.0        0.500E+00  0.171E+01  0.312E+01  0.440E+01  0.550E+01
    0.679E+01  0.770E+01  0.858E+01  0.990E+01  0.110E+02  0.121E+02
    0.130E+02  0.143E+02  0.154E+02  0.165E+02  0.177E+02  0.187E+02
    0.198E+02  0.209E+02  0.220E+02

SCF3

DERIVED PARAMETER:
BLADE STATIC PRESSURE COEFF
DOUBLEROW - BOTH             COUNTER -    611
1ST DIMENSION -     TIME (SECONDS)       256 POINTS  0.186 SECONDS
2ND DIMENSION (ROW POSITION) - FRACTN OF CHORD
    0.999E-02  0.300E-01  0.799E-01  0.150E+00  0.200E+00  0.250E+00
    0.350E+00  0.400E+00  0.450E+00  0.500E+00  0.550E+00  0.599E+00
    0.699E+00  0.919E+00
3RD DIMENSION (COLUMN POSITION) - FRACTN OF RADIUS
    0.400E+00  0.750E+00  0.864E+00  0.955E+00
```

Figure 35. Sample menu of scratch file contents.

```
MENU/INFO/
 ----------MENU LISTING OF INFO FILE------------------------------

 INITIAL GROUP:
  MRAZ R992 338., R106 0.0, R018 30.32/
  TRAZ R025 45.0/
  MDEG A320 0.0/
  TDEG A333 0.0/
  TIAS P002/
  TASK A173/
  OATM T004/
  STAT P030/
  MTOR M107/
  END

 SUBSEQUENT GROUPS IN FILE:
  S1BV  BLADE BEAMWISE VIBRATION
  S1CV  BLADE CHORDWISE VIBRATION
  S2BU  BL BUTTONS   UPPER SURFACE
  S2BL  BL BUTTONS   LOWER SURFACE
  S2HW  HOT-WIRE ATTENUATION SENSORS
  S1BB  BLADE BEAMWISE BENDING
  S1CB  BLADE CHORDWISE BENDING
  S1BT  BLADE TORSION
  S2HS  HOT-WIRE ATTENUATION - SPECIAL
  S2PP  BLADE ABSOLUTE PRESSURE
  S2PX  BLADE ABSOLUTE PRESSURE
  BBM1  RTR 1, BLD 1, BEAM BENDING MOMENTS
  CBM1  RTR 1, BLD 1, CHORD BENDING MOMENTS
  TBM1  RTR 1, BLD 1, TORSIONAL BENDING MOMENTS
  CLR1  RTR 1, BLD 1, TOTAL LIFT COEFFICIENT
  CDR1  RTR 1, BLD 1, DRAG COEFFICIENT
  CMR1  RTR 1, BLD 1, TOTAL PITCH MOMENT COEF
  CNR1  RTR 1, BLD 1, NORMAL FORCE COEFFICIENT
  CCR1  RTR 1, BLD 1, CHORDWISE FORCE COEF
 END-OF-FILE ON INFO FILE AFTER  345 LINES READ

 NEW STEP  STEP CPU  0.52 SEC  TOTAL CPU   3.0 SEC
```

10
B

Figure 36. Sample Info File menu.

```
 NEW STEP  STEP CPU  0.01 SEC  TOTAL CPU   2.4 SEC
MENU/EDIT/

    COMMAND SEQUENCE FILE
    MAX NUMBER OF COMMAND SEQUENCE
    BLOCK NAMES ALLOWED IS   6
    EXISTING BLOCK NAMES ARE:
         APL1
         CNPL
         CCCC
         DDDD
         EEEE
         FFFF


 NEW STEP  STEP CPU  0.02 SEC  TOTAL CPU   2.4 SEC
MENU/MASK/

     LIST OF MASKED ITEM CODES

 B123  B124  B125  B126  B127  B128
 B129  B130

        8 ITEM CODES LISTED
```

Figure 37. Sample Command Sequence block and masked item code menus.

```
-------------CURRENT RUN SETTINGS---------------
SETTINGS THAT THE 'SET/ //' COMMAND CAN CHANGE:

                      SETTING   ---KEYWORDS---
  ROTOR MODE          'MAIN'    'MAIN','TAIL'
  SCREEN PLOT SIZE    'FULL'    'FULL','HALF'
  PLOT GRID MODE      'GRID'    'GRID','NOGRID'
  PLOT TICS MODE      'NOTICS'  'TICS','NOTICS'
  PLOT LABEL SPEED    'SLOW'    'QUICK','SLOW'
  PLOT COPY MODE      'NOCOPY'  'COPY','NOCOPY'

SETTINGS THAT CANNOT BE CHANGED IN THIS RUN:

  TERMINAL DATA RATE                  240 CHARS/SEC
  PLOT FRAME WIDTH                   8.50 INCHES
  INITIAL PEN PLACEMENT IN 'X'      -1.50 INCHES
  INITIAL PEN PLACEMENT IN 'Y',      0.50 INCHES
  PRINT BLOCKS, 5 LINES/BLOCK           6 PER PAGE
  SCRATCH FILES SIZE                  225 RECORDS
  CPU TIME TO TRIGGER WARNING         90. SECONDS
  STEP EXECUTION TIMES       WILL NOT BE PRINTED

NEW STEP.
```

Figure 38. Sample run setting menu.

the i'th entry following the block name in the execution command is the i'th parameter and will be inserted wherever an entry "%i" is found in the command sequence. Each parameter entry must begin with the "%" character but this character is dropped before insertion in the command sequence. Parameters are always interpreted as strings in the execution commands before they are inserted in the command sequence. Following insertion, numbers are reinterpreted. Parameters may contain no more than 20 characters excluding the "%" character. If a parameter is not required, then it should be allowed to default.

Source - Source of data for processing. This entry specifies whether input data will come from the Master File or a scratch file. If data come from the Master File, the entry can be either an item code or 'GROUP'. The entry 'GROUP' specifies that an Info File group will be used to define the item codes for input. If data are to come from a scratch file, the user should enter "SCF1", "SCF2" or "SCF3" as appropriate. There is no default for this entry.

(Time) - Time offset in seconds. This entry can have two slightly different meanings. When the input record length is defined with (Duratn), then (Time) is the offset from the start of data on the Master File to the start of data in the input record. When the input record is defined with (Cycles), then (Time) is still an offset from the start of data on the Master File. However, in this case the input data record will start at the beginning of the first complete rotor cycle after this offset. The default for this entry is zero seconds.

1ST DIM - Definition for the domain of the first dimension variable. When data are extracted from a scratch file, the user has an option. Each data stream to be retrieved may have all available data points extracted or only a single data point may be retrieved from each stream. The keyword 'ALL' specifies that all data points will be retrieved. A numeric entry specifies a single data point. The meaning of the number is dependent upon the subsequent entry (1st Var) for a variable definition. The

purpose of this option is to allow the user to select a particular azimuth value, time, frequency, or harmonic in retrieving data from a scratch file. The default for this entry is 'ALL'.

(2nd Dim) - Selection for the domain of the second dimension variable. When data are extracted from a scratch file or from the Master File using an Info File group, the user has an option regarding the second dimension, which is chord position in the OLS blade example. Data streams corresponding to all second independent variable positions available are retrieved when the user specifies 'ALL'. Alternatively, the user may specify an element number corresponding to a particular second dimension position for which data streams are available. Element numbers correspond to row sequence positions stored on the specified group of the Info file or on the scratch file. Allowed element numbers are 1 through 64. 'ALL' is the default for this entry.

(3rd Dim) - Selection for the domain of the third dimension which is radial position for the OLS blade example. This entry option is the same as the second dimension option above except that an element should correspond to a stored column element number. 'ALL' is the default for this entry.

(1st Var) - Variable definition for a numeric **1st Dim** entry. The meaning of a numeric **1st Dim** entry is established by (1st Var). Data may be stored on the scratch file as a function of time in seconds, frequency in Hz, or Harmonic Number. If the data are stored as a function of Harmonic Number, then 'HARM' or 'IMPL' may be entered to indicate that **1st Dim** specifies a Harmonic Number. If the data are stored as a function of frequency, then 'FREQ' or 'IMPL' may be entered to indicate that **1st Dim** specifies a frequency. When the data are stored as a function of time, then more options are available to the user. 'TIME' or 'IMPL' may be entered to indicate that **1st Dim** specifies a time measured from the beginning of data stored on the Master File partition from which the data were retrieved. Alternatively,

'MRAZ', 'TAS' or 'MRPM' may be entered to indicate that **1st Dim** is an azimuth position, true airspeed, or rotor rpm, respectively. 'IMPL' is the default for this entry.

Consider an example of the use of Table 9. Suppose a user wishes to derive C_p from absolute pressure data stored on scratch file one for a rotor azimuth position of 90 degrees, for all available chord and radial stations on the top surface of the blade. Assume that the Specification and Action substeps are already entered:

DERI/CP 264 CALC 14.35/

The user can find the appropriate Input substep sequence for DERI/CP in Table 9. The first Input Substep entry is indicated by the descriptor **SOURCE** where a branch occurs in the allowable sequences. The possible entries are listed below this descriptor. (Item Code) is an entry descriptor defined in the **Source** explanation. 'GROUP,' 'SCF1,' 'SCF2', and 'SCF3' are actual possible entries also defined in the **Source** explanation. Since the input data are stored on scratch file one, the user should enter 'SCF1' for the first entry.

Proceeding to the right from the selected entry, a closed bracket unites the paths from each of the scratch file selections. Then an open bracket shows that the paths branch for the ***1ST DIM** entry descriptor*. Underneath this descriptor, the descriptor (Value) and the possible entry 'ALL' are listed. These possible entries are defined in the **1st Dim** explanation. In particular, the user wishes to select the single instant of 90 degrees of azimuth so a (Value) of 90 is entered. Proceeding directly to the right of the (Value) descriptor, the descriptor for the next entry is (1st Var), which should define the meaning of the number 90. The possible entries are not listed under (1st Var) since no branch occurs at this point in the sequence. From the explanation for (1st Var), the proper entry is 'MRAZ' to indicate that 90 is rotor azimuth in degrees.

Again proceeding to the right, the final three entries involve no branches. From the explanation for the descriptors (2nd Dim), (3rd Dim), and (Dblrow), the appropriate entries are 'ALL,' 'ALL,' and 'TOP.' Thus the command developed through the Input substep would be:

DERI/CP 264 CALC 14.35/

SCF1 90 MRAZ ALL ALL TOP/...

with the Disposition substep still to be entered.

As a second example, consider the execution of a command sequence. Assume that the following command sequence is stored in the command sequence storage file with the block name "AAAA."

```
ANAL/AVER/GROUP S2PP BOTH ALL ALL %1 0 3/KEEP SCF1/

DERI/CP 264 CALC %2/SCF1/KEEP SCF2/

DERI/CN/SCF2/KEEP SCF3/

COMM/LEVEL FLIGHT AT %3 KNOTS/

DISP/SCF3/CONT CYL MRAZ IMPL/

NOEDIT/
```

The "%" entries must be replaced during execution of the sequence by parameters. As shown in Table 9, the EXECUTE command has no branches. The first entry in the Input Substep is the (BLOCK NAME), which is "AAAA" for this example. The parameters follow with each parameter number specified by position following the block name. Thus, parameter one is the counter for the original pressure data, parameter two is the static pressure for the C_p derivation, and parameter three is the airspeed. Assuming the counter is 870, the static pressure is 14.3 PSIA, and the airspeed is 50 knots, then the execution command is:

```
EXECUTE/AAAA %870 %14.3 %50/
```

Notice that the "%" character must precede each parameter entry. Since parameter four does not appear in block "AAAA", this parameter is not included in the execution command and defaults to "no entry." Following are the commands that will be processed following the above execution command.

```
ANAL/AVER/GROUP S2PP BOTH ALL ALL 870 0 3/KEEP SCF1/

DERI/CP 264 CALC 14.3/SCF1/KEEP SCF2/

DERI/CN/SCF2/KEEP SCF3/

COMM/LEVEL FLIGHT AT 50 KNOTS/

DISP/SCF3/CONT CYL MRAZ IMPL/

NOEDIT/
```

5.5 DISPOSITION SUBSTEP COMMANDS

The available sequences of Disposition Substep entries are the same for all possible combinations of Specification, Action, and Input Substeps, provided that a Disposition Substep is required. The user must assume the responsibility to assure that the selected Disposition can be performed given the specified Specification, Action, and Input. When an impossible Disposition is entered, the command step will be accepted for execution and then an error message generated in the processing part of the program. For example, if the user's Specification, Action, and Input substeps result in a process that creates a single time history output and the Disposition calls for a CONTOUR plot, the command step will be accepted but an error message will be generated in processing and no output will appear.

The Disposition Substep sequences are displayed in Table 10 using the same format as Table 9. However, Specification, Action, and Input Substep combinations are not listed since the Disposition Substep has the same available options for each combination that requires a Disposition.

The meaning for every Disposition Substep entry descriptor and possible inputs for each are given below:

CROSS - Tektronix cross-hair cursor activation. This entry determines whether the Tektronix cross-hair cursor will be activated after an X-Y plot is completed to evaluate points on the plot in user coordinates. 'CLOSE' indicates the cursor should not be activated. 'CURSOR' indicates the cursor should be activated. A 'CURSOR' request is ignored in the batch operating mode. 'CLOSE' is the default for this entry.

(DECADES) - Decades in log scale. (DECADES) is an integer that specifies the number of decades that will be depicted for the logarithmic scale in question. One to twelve decades are allowed in the X (independent variable) or Y (dependent variable) direction. The default number of decades is three in either direction.

FORMAT - Overall format for the output. This entry specifies the type of output. The allowed formats are:

PLOT - X-Y plot with a single curve (Figure 25)

TABLE 10. DISPOSITION SUBSTEP SEQUENCES.

```
         **INDEP**
          IMPLIED
          TIME
PLOT      FREQUENCY   **CROSS**   **Y-INTERVAL**               **X-INTERVAL**
MPLOT     HARMONIC      CURSOR      AUTO      AUTO               AUTO      AUTO/
LPLOT     TAS           CLOSE       (VALUE)   (VALUE)            (VALUE)   (VALUE)/
          MRAZ                      LOG       (DECADES)          LOG       (DECADES)/
          MRPM
          ROW
          COLUMN

         **INDEP**
          IMPLIED
          TIME                      **Y-INTERVAL 1**             **Y-INTERVAL 2**
          FREQUENCY   **CROSS**     AUTO      AUTO               AUTO      AUTO
DPLOT     HARMONIC      CURSOR      (VALUE)   (VALUE)            (VALUE)   (VALUE)
          TAS           CLOSE       LOG       (DECADES)          LOG       (DECADES)
          MRAZ
          MRPM
          ROW
          COLUMN

    --> **X-INTERVAL**
          AUTO      AUTO/
          (VALUE)   (VALUE)/
          LOG       (DECADES)/

                  **X-VAR**
                   IMPLIED       **Y-VAR**
          **SYSTEM**  TIME         IMPLIED
CONTOUR     RECT     FREQUENCY     TAS       (Z-INT) (Z-VAL) (Z-LEVLS)/
            CYL      HARMONIC      MRAZ
                     TAS           MRPM
                     MRAZ          ROW
                     MRPM          COLUMN
                     ROW

                  **X-VAR**
                   IMPLIED       **Y-VAR**
          **SYSTM**  TIME          IMPLIED
SURFACE     RECT     FREQUENCY     TAS       (OBS POS X) (OBS POS Y)
            CYL      HARMONIC      MRAZ
                     TAS           MRPM
                     MRAZ          ROW
                     MRPM          COLUMN
                     ROW

    --> (OBS POS Z)/

KEEP      SCF1
ADD       SCF2    (USER VAR)/
          SCF3
```

MPLOT	-	X-Y plot with one or more curves (Figure 1)
LPLOT	-	X-Y comparison plot to which curves can be added using APLOT with extensive annotation for each curve (Figure 12)
DPLOT	-	X-Y plot with separate scaled areas for each double-row element (Figures 7 and 21)
APLOT	-	Add a curve to an existing multiple curve X-Y plot that was initially drawn using MPLOT, LPLOT, or DPLOT (Figures 12 and 27)
PRINT	-	Print output (Figure 19)
CONTOUR	-	Contour plot of a function of two independent variables (Figures 2 and 5)
SURFACE	-	Surface plot of a function of two independent variables (Figures 3 and 4)
KEEP	-	Store the process results on the designated scratch file while destroying any data already present on the file.
ADD	-	Store the process results on the designated scratch file in addition to data already present on the file.

There is no default for this entry.

INDEP - Independent scale variable for X-Y plot. The basic independent scale variable for an X-Y plot will be completely determined by the input and process performed in generating the output function. Sometimes, however, the computer may hold sufficient information to display associated scale variables for a given dimension and the user may choose between these variables. For example, the user must frequently choose between time or azimuth as the first dimension scale variable.

When a single curve is to be plotted, the data must be a function of either the first, second, or third dimension and the independent scale variable specified must be associated with this dimension. When more than one curve is to be plotted in a single command step (i.e., using MPLOT or DPLOT), the data are a function of two of the three dimensions and the independent variable specified must be associated with the numerically lower dimension present. For example, if the data are a function of row and column (e.g., chord and radius), then row (chord) must be the independent scale variable, while each curve on the multiple curve X-Y plot represents a different column element (radius position). If the data are a function of the first dimension, then this dimension must be time, harmonic number, or frequency. For frequency, 'FREQ' or 'IMPL' may be entered to indicate a frequency scale. For harmonic number, 'HARM' or 'IMPL' may be entered to indicate a harmonic number scale. For time, 'TIME' or 'IMPL' may be entered for a time scale or 'MRAZ', 'TAS', or 'MRPM' may be entered to specify rotor azimuth, true airspeed, or rpm scales, respectively. If the second dimension is the independent variable dimension, then 'ROW' or 'IMPL' may be entered to indicate a chord position scale. If the 3rd dimension is the independent variable dimension, then this dimension may represent geometric position (radius) or multiple counters. When this dimension represents position or when the dimension is multiple counters with a user-entered number for each counter to be used as a scale, 'COLUMN' or 'IMPL' may be entered. When this dimension represents multiple counters, the user also has the option to enter

'MRAZ', 'TAS', or 'MRPM' to select scales of rotor azimuth, true airspeed, or rotor RPM, respectively. 'IMPL' is always the default for this entry.

(OBS POS X) - Observer position in the X direction. For SURFACE format plots using rectangular coordinate systems, this entry specifies the observation point for the plot in the first independent variable or X scale. For surface format plots using cylindrical coordinate systems, the X axis is interpreted as the zero-degree direction on the Z (dependent variable) = 0.0 plane. The units of this entry are widths of the displayed function in the X direction. For rectangular coordinate system plots, the X or Y scale is adjusted so that the apparent width of the function in the X direction is approximately the same as the width in the Y direction. The allowed values for this entry are from 1000 to -1000. The default value is 10 for both cylindrical and rectangular coordinate systems.

(OBS POS Y) - Observer position in the Y direction. This entry is the same as (OBS POS X), except that position in the second independent variable or Y direction is given. For cylindrical systems, this direction corresponds to 90 degrees. Allowed values are the same as for (OBS POS X). The default value is 10 for rectangular systems and .3 for cylindrical systems.

(OBS POS Z) - Observer position in the Z direction. This entry is the same as (OBS POS X), except that it indicates position in the dependent variable or Z direction. Allowed values are the same as for (OBS POS X). The default value is 10 for both rectangular and cylindrical coordinate systems.

For observer position specifications in SURFACE plots, the user must assure that the observer position is not between two parts of the surface.

SCRATCH - Scratch file for data storage. This entry specifies the scratch file that will hold the process output as specified by a 'KEEP' or

'ADD' entry. The allowed file names are 'SCF1', 'SCF2', or SCF3'. There is no default for this entry.

SYSTEM - Coordinate system for three-dimensional plot. This entry specifies the coordinate system to be used in generating a SURFACE or CONTOUR plot. The allowed entries are 'CYLINDRICAL' and 'RECTANGULAR'. The default for this entry is 'CYLI'.

(USER VAR) - User supplied value to position scratch file columns on the third dimension independent variable scale. This entry provides the user the capability to generate a scale in the column position direction. For example, the Master File partition might contain several counters representing different rates of descent. The user could process data from each counter and store the results each time using the 'ADD' instruction. Along with each 'ADD', the user could supply the rate of descent for the entry (USER VAR). These rates of descent could then be used to position the corresponding column element outputs on a rate of descent scale. Allowed inputs for this entry are 'NONE' or numbers ranging from -10000000 to +10000000. The default for this entry is 'NONE'.

X-BOT - Minimum X scale value. This entry specifies the minimum independent variable or X scale value to depict for a single or multiple curve X-Y plot. The entry 'AUTO' specifies automatic scaling to select this value. A numeric input specifies a user-selected value. If this value is specified in conjunction with a user-specified X-interval, then **X-BOT** must be an integer multiple of **X-INT**. Allowed input values for **X-BOT** are 'AUTO' or numeric values between -10000000 and +10000000. The default for this entry is 'AUTO'.

X-INT - X scale interval. This entry specifies the interval between annotated positions on the independent variable axis in independent variable units. The entry 'AUTO' specifies automatic scaling for this value. A numeric input specifies a user-selected value. This number should be a power of ten times one of the

numbers: one, two, four, or five. Allowed input values for **X-INT** are 'AUTO' or numeric values between 0.0 and 10,000. The default for this entry is 'AUTO'.

X-VAR - X Direction variable. For surface or contour plots, this entry specifies the variable for the first independent axis. This axis could either be 2nd dimension position (which is chord for the OLS blade example), or a time-related variable. The allowed display axis variables are dependent upon the input and process steps. If the data to be plotted is a function of the 2nd and 3rd dimensions, then **X-VAR** must be the 2nd dimension which can be called 'ROW' or 'IMPL'. If the data to be plotted is a function of the 2nd or 3rd dimensions and the 1st dimension, then **X-VAR** must be time related. If the 1st dimension variable is frequency, then the allowed entries for **X-VAR** are 'FREQ' or 'IMPL', which both specify frequency. If the time related variable is a harmonic number, then the allowed entries are 'HARM' or 'IMPL', which both specify harmonic number. If the time related variable is time, then allowed entries are 'TIME', 'IMPL', 'MRAZ', 'TAS', and 'MRPM'. 'TIME' and 'IMPL' both specify time. 'MRAZ', 'TAS', and 'MRPM' specify rotor azimuth, true airspeed, and rotor speed, respectively. 'IMPL' is always an allowed entry and is the default for this entry.

Y-BOT - Minimum Y scale value. This is the corresponding entry for the dependent variable plot axis as **X-BOT** is for the independent variable plot axis.

Y-BOT n - Min Y scale values for DPLOT option. "n" can be "1" or "2". Definition and options are the same as for **Y-BOT** except that DPLOT requires two scale specifications. "**Y-BOT 1**" applies to the TOP double-row element, which will appear at the top of the plot frame. "**Y-BOT 2**" applies to the BOTTOM double-row element, which will appear lower in the plot frame.

Y-INT - Y scale interval. This is the corresponding entry for the dependent variable plot axis as **X-INT** is for the independent variable plot axis.

Y-INT n - Y scale intervals for DPLOT option. "n" can be "1" or "2". Definition and options are the same as for **Y-INT** except the DPLOT requires two scale specifications. "**Y-INT 1** applies to the TOP double-row element, which will appear at the top of the plot frame. "**YINT 2**" applies to the BOTTOM double-row element, which will appear lower in the plot frame.

Y-VAR - Y Direction variable. For surface or contour plots, this entry specifies the variable for the second independent axis. This axis could be the 2nd dimension position (chord in the OLS blade example) or the 3rd dimension position. The 3rd dimension (column) can be a radial station in the OLS blade example, or a variable created by one or more ADD's to a scratch file. If the data to be plotted is a function of the 2nd and 1st dimension, then (Y-Var) must be the 2nd dimension variable which can be called 'ROW' or 'IMPL'. If the data to be plotted is a function of the 3rd dimension and the 2nd or 1st dimension, then (Y-Var) must be associated with the 3rd dimension (column position). If the 3rd dimension variable is geometric or if the user has entered column position values in writing the column positions separately to a scratch file, then the appropriate entries are 'COLUMN' or 'IMPL', where the scale will be the column positions specified by the Info File or the user. If the 3rd dimension variable has been generated by storing multiple counters on a scratch file to generate a variation in airspeed or rotor speed, then the appropriate (Y-Var) entries are 'TAS', 'MRPM', or 'MRAZ' for true airspeed, rotor speed, or rotor azimuth scales, respectively. 'IMPL' is the default for this entry. However, if the 3rd dimension is formed from several ADD's to a scratch file and no user-supplied numbers have been entered as column positions, then the entry 'IMPL' will create an error condition in processing.

Z-INT - Interval between contour levels. This entry allows the user to specify the interval between levels depicted by a contour plot. These levels correspond to dependent variable or Z values held constant for each individual contour drawn. The interval must be some power of ten times one, two, four, or five. The alternative entry is 'AUTO', which selects autoscaling. 'AUTO' is the default for this entry.

(Z-LEVLS) - Maximum number of levels for a contour plot. If (Z-INT) and (Z-VAL) are specified directly, the program will attempt to draw (Z-LEVLS) contour levels beginning with (Z-VAL) and incrementing by (Z-INT). If (Z-INT) and (Z-VAL) are specified as 'AUTO', the program will optimize the contour level spacing between the minimum and maximum dependent variable values that occur. In either case, fewer than the maximum number of contour levels may actually be drawn. Sixteen is the default for this entry until another number is specified. Then the new number becomes the default. As many as 32 contour levels will be properly annotated. More contour levels may be selected but will not be properly annotated.

Z-VAL - Specified contour value. This entry allows the user to enter the minimum contour level for a contour plot. This value must be an integer multiple of the Z-interval or 'AUTO' for autoscaling may be specified. If a specified numeric value is not within the range of the function, then the corresponding contour will not appear in the plot but the level will be labeled in the annotation area. 'AUTO' is the default for this entry.

5.6 USER INPUT DURING COMMAND STEP EXECUTION

Normally, all user instructions are entered in the form of command steps before the execution of those steps, and no additional instructions are entered until the step execution is complete. Occasionally, however, the user must enter instructions during the execution of a step. These occasions are restricted to the Interactive and Interactive Graphics modes of operations.

5.6.1 The 'Change' Mode of the 'EDIT' Specification

The EDIT/CHANGE command provides the capability to perform the following operations on a recorded sequence, or block, of command steps: list the sequence, change lines in the sequence, insert lines in the sequence, and delete lines from the sequence. Certain other functions are provided in this mode to facilitate use of these capabilities. These functions include renumbering the block of command steps, listing available commands, listing certain lines from the block, and exiting the Change mode.

When the Change mode is entered, a completely new structure and syntax for user instructions is introduced. Six keyword commands are recognized: $A, $C, $L, $N, $E and $?. Some of these keywords may be followed by one or two line numbers. Line numbers may be determined from a listing ($L) of the block. Following is a list of keyword commands with '( )' used to signify possible line number entries. The numbers themselves should not be enclosed in parentheses.

$A ( ) - Insert one or more lines. When this command is entered, the user may enter one or more lines under the command line. These lines will be inserted just after the line specified by ( ). The insertion lines end when a command keyword is entered.

$C ( ) ( ) - Change or delete lines. When this command is entered, one or more lines are deleted and zero or more lines may be entered to replace them. If one line number is specified, then that line is deleted. If two or more line numbers are specified, then those lines are deleted along with any lines that lie between the specified lines. The second line number specified must be greater than the first. If any lines are entered underneath the $C command, they will be inserted between the lines that preceded and followed the group of lines that was deleted. The insertion lines end when a command keyword is entered.

$L ( ) ( ) - List lines or whole block. This command causes one or more lines to be listed along with the corresponding line numbers. When no line numbers are included in the command, the entire block is listed. If a single line number is included, the corresponding line and all subsequent lines in the block are listed.

When two line numbers are entered with the command, the two corresponding lines and any intervening lines are listed. The second line number specified must be greater than or equal to the first. When both line numbers specified are the same, then a single line is listed. If a listing is specified after changes or insertions have been made but before a renumber operation ($N), then the inserted lines will be listed without the line numbers. The line numbers displayed in a listing are valid until a renumber is performed on the block.

$N - Renumber the block. This command causes new line numbers to be assigned to each line so that every line is numbered. Lines inserted under the '$A' or '$C' commands will be unnumbered until a $N command is used. The $N command also removes any restriction on the line numbers that can be addressed by the $C and $A commands. The $C and $A commands must always address lines sequentially until an intervening $N command is used. For example, if '$A 30' is entered, then '$C 10 20' is an illegal command until a '$N' command is used.

$E - End change mode. This command causes the
change mode to exit. When '$E' is entered, the 11
following message is printed. B

ENTER STORE OR KILL

The user should enter 'STORE' or 'KILL' as appropriate. If 'STORE' is entered, the block specified by name when the 'EDIT/CHANGE' mode was entered is replaced by the modified block just created. If 'KILL' is entered, the specified block is unchanged and the modifications specified under the 'EDIT/CHANGE' mode are discarded. After 'STORE' or 'KILL' is entered, the computer returns to normal user command step input mode.

$? - List available commands. This command causes the computer to list all six available commands along with brief descriptions of each. The same listing is provided automatically when the 'EDIT/CHANGE' mode is first entered.

A command keyword for the EDIT/CHANGE mode must be the first entry on a line. The following two entries may be line numbers as needed. Two entries, keyword or numeric, should be separated by a comma or one or more blanks. Only one command keyword may appear on a line. Lines for insertion should not be included on the same line as command keywords. The inserted lines will be copied exactly as entered, without error checking.

5.6.2 Tektronix Cross-Hair Cursor

When a Disposition Substep specifies that the Tektronix Graphics Cursor shall be activated, the cursor cross-hairs appear on the screen following completion of the plot. The user can manipulate the Graphic Cursor by using the thumb-wheels on the right-hand side of the Tektronix keyboard. When the cross-hair intersection is at a point on the plot that the user wishes evaluated, any key on the keyboard can be struck to extract the coordinates of the point in the scale of the plot. The coordinate values are printed on the left-hand side of the screen.

The coordinate values printed represent the position of the cross-hair cursor on the plot. The user must assure that the cursor intersection is actually located on the plot point that is to be evaluated. The Tektronix screen consists of a raster matrix of points that may be addressed. The raster separation limits the accuracy of both the plot itself and the graphic cursor evaluation of points on the plot. The computer indicates this potential inaccuracy by calculating a number equivalent to one-half the raster separation in the scale of the plot. This number is printed for both the dependent and independent variables just to the right of the corresponding scale value. A typical Graphic Cursor point evaluation is printed:

```
0.9054E-01 +/- .94E-03     0.1476E+02 +/- .94E-03
```

The left-hand values are the independent variable location and accuracy, respectively, and the right-hand values are the dependent variable location and accuracy.

If the user enters the character 'C' to request a point evaluation, then the Graphics Cursor will remain active after the location has been printed so that the cursor may be repositioned for additional point evaluations. When any character except 'C' is entered, then the Graphics Cursor is deactivated after the current location is evaluated and the computer will proceed to the next command step.

Point locations may be evaluated from log-log or semilog plot scales as well as from linear scales. The user may notice that the computed accuracy of the point evaluation changes dramatically as cursor position is changed on a log scale.

5.6.3 Printout Control

When printed output is generated in the Interactive or Interactive Graphics mode, the printout is written a page at a time. At the end of each page, the user is informed of the independent variable interval included on the next page and given the option to print the next page, skip the next page, or skip the balance of the printout for the current row and column position. An example of the message that gives this information and option is:

```
NEXT PAGE .9375E+00 TO .9961E+00 ENTER:C=CONTINUE,S=SKIP,X=STOP
```

This message was compressed slightly to fit on this page.

If the user enters 'C', the next page will be printed and, if the printout is incomplete, the above message will be repeated with the interval changed to correspond to the subsequent page. If the user enters 'S', the next page will not be printed and if the printout is incomplete, the above message will be repeated for the subsequent page. If the user enters 'X', the balance of the values for the current row and column position are skipped and printout begins for the next row/column position. If the last row and column are currently being printed, or if there is a single data stream being printed, then the computer proceeds to the next command step.

5.6.4 Menu Listing Control

Menu listing control is very similar to printout control. When a menu of item codes or counters is generated, the menu is printed in 50-line blocks. Frequently, all of the item codes or counters will fit in one block. However, if multiple blocks are required, then a message similar to the following example will appear at the end of each block.

```
NEXT BLOCK P985 TO T004  ENTER: C=CONT, S=SKIP, X=STOP
```

The meaning of these control entries is the same as the printout control entries except that the entry 'X' will always cause the computer to proceed to the next command step.

5.7 EXAMPLES OF COMMAND STEPS

The purpose of this section is to give the user some examples of command steps and sequences of command steps to illustrate

the structure, syntax, and allowed entries for user input. In addition, the examples will show basic approaches to typical problems that are handled with DATAMAP. Explanations for each command step precede the commands. The commands form sequences that might represent typical runs using the Processing Program in the Interactive Graphics and Batch Modes. In each example it is assumed that the required data have already been transferred to the Master File and that the user has already been through the Initialization Phase of program execution (see Section 5.1).

Two conventions are followed to enhance the clarity in the examples of this Section and in examples from other Sections. First, all computer messages and user command input are typed in upper case while explanatory text in the report is typed in normal mixed upper and lower case. Second, user command input always begins at the left-hand margin of the page while computer messages always begin at least one space in from the left-hand page margin. This second convention agrees with the actual interaction convention used for interactive program operation.

5.7.1 Examination of Blade Pressure Data

Blade absolute pressure data have been loaded in a partition of the Master File. Data from each of the sensors on the OLS blade are present for six different steady state flight conditions (i.e., level flight for six different airspeeds). The uniform sample rate for these data is 2048 samples/second. Additional data present on the partition are True Indicated Airspeed, Boom System Static Pressure, Outside Air Temperature, Azimuth, and Main Rotor Mast Torque. The user enters the Processing Program in the Interactive Graphics mode and selects the partition, 'C81COMPR', for access. Immediately after the Initialization Phase, the user requests a listing of the counters on the partition, which represent the various flight conditions. The command is entered

```
 NEW STEP
MENU/DATA/
```

and the counters 610, 611, 612, 613, 614, and 615 are listed on the Tektronix screen.

Next, the user wishes to derive True Airspeed for each of the six counters. Since the airspeed is nearly constant and may intersect a plot grid line, the user selects the NOGRID mode.

```
 NEW STEP
SET/NOGRID/
```

Then the user derives True Airspeed for counter 615. Calibration constants representing the slope and Y-intercept of a line used to convert true indicated airspeed to calibrated airspeed (see Section 6.2) must be entered.

```
 NEW STEP
DERIVE/TAS,,,1.0208 3.33334/615 0 6/PLOT/
 EXECUTING
```

Figure 39 shows the True Airspeed plot that is generated by this command. The True Airspeed is essentially constant at 129 knots. This same operation is repeated for the other counters although the commands and resultant plots are not shown. Once the calibration constants, 1.0208 and 3.33334, have been entered, they become the default constants and need not be reentered for the balance of the program run. The derived True Airspeeds for all six counters are:

610 - 143 knots

611 - 72 knots

612 - 86 knots

613 - 101 knots

614 - 115 knots

615 - 129 knots

The user decides to process counter 615 first. A representative rotor cycle of data for each absolute pressure sensor on the blade is computed.

```
 NEW STEP
ANALYZE/AVERAGE/GROUP S2PP,,,,615 0 5/KEEP SCF1/
 EXECUTING
```

The above command selects cycle averaging of five contiguous rotor cycles from each of the item codes that are identified in Info File group S2PP. The three default entries specified by the sequence of four commas are BOTH double-row elements (i.e., upper and lower blade surface), ALL column elements (i.e., all span stations), and ALL row elements (i.e., all chord stations). The first rotor cycle in the sequence of five that are processed is the first complete rotor cycle after the start of data. The results of this operation are stored in scratch file one and no graphic or printed output is generated from this command step.

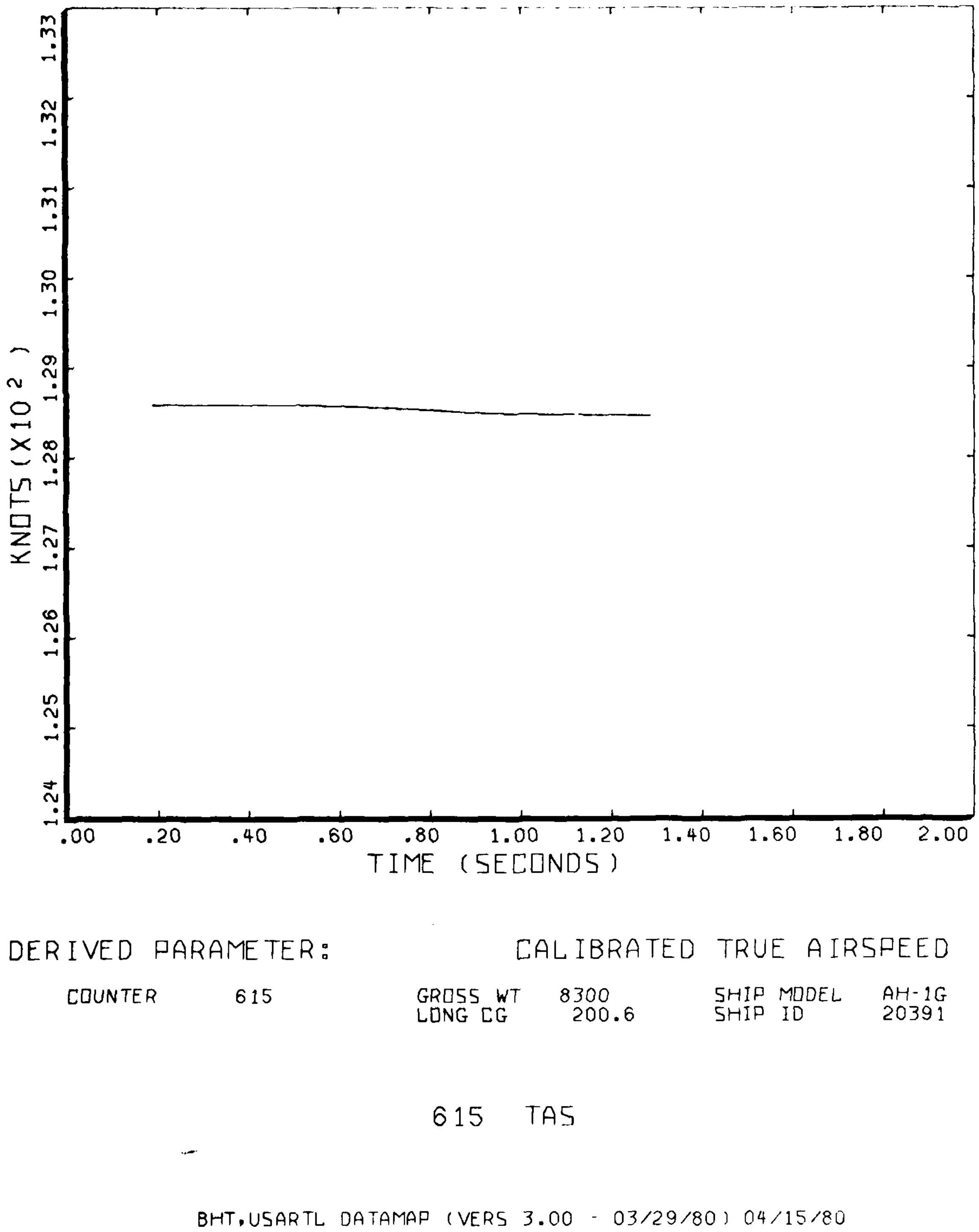


Figure 39. True airspeed plot with NOGRID option.

For a preliminary look at the results of this operation, the user displays, as a function of blade azimuth, the pressure data for the fourth column element (86 percent span station) on the top surface of the blade. First, the user resets the plot mode to GRID and enters an appropriate comment for subsequent plot labeling.

```
 NEW STEP
SET/GRID/

 NEW STEP
COMM/LEVEL FLIGHT AT 129 KNOTS  OLS DATA/

 NEW STEP
DISPLAY/SCF1 ALL ALL 4 TOP/MPLOT MRAZ/
 EXECUTING
```

Figure 40 is the resultant plot from this command. The curves in this graph show substantial high-frequency components. It cannot be assumed that these components are noise. However, these components are outside the frequency band of interest for this particular example. In addition, the high-frequency components make the dash-dot line encoding for multiple curve plots very difficult to read. Accordingly, the data in scratch file one are digitally filtered and the results stored on scratch file two.

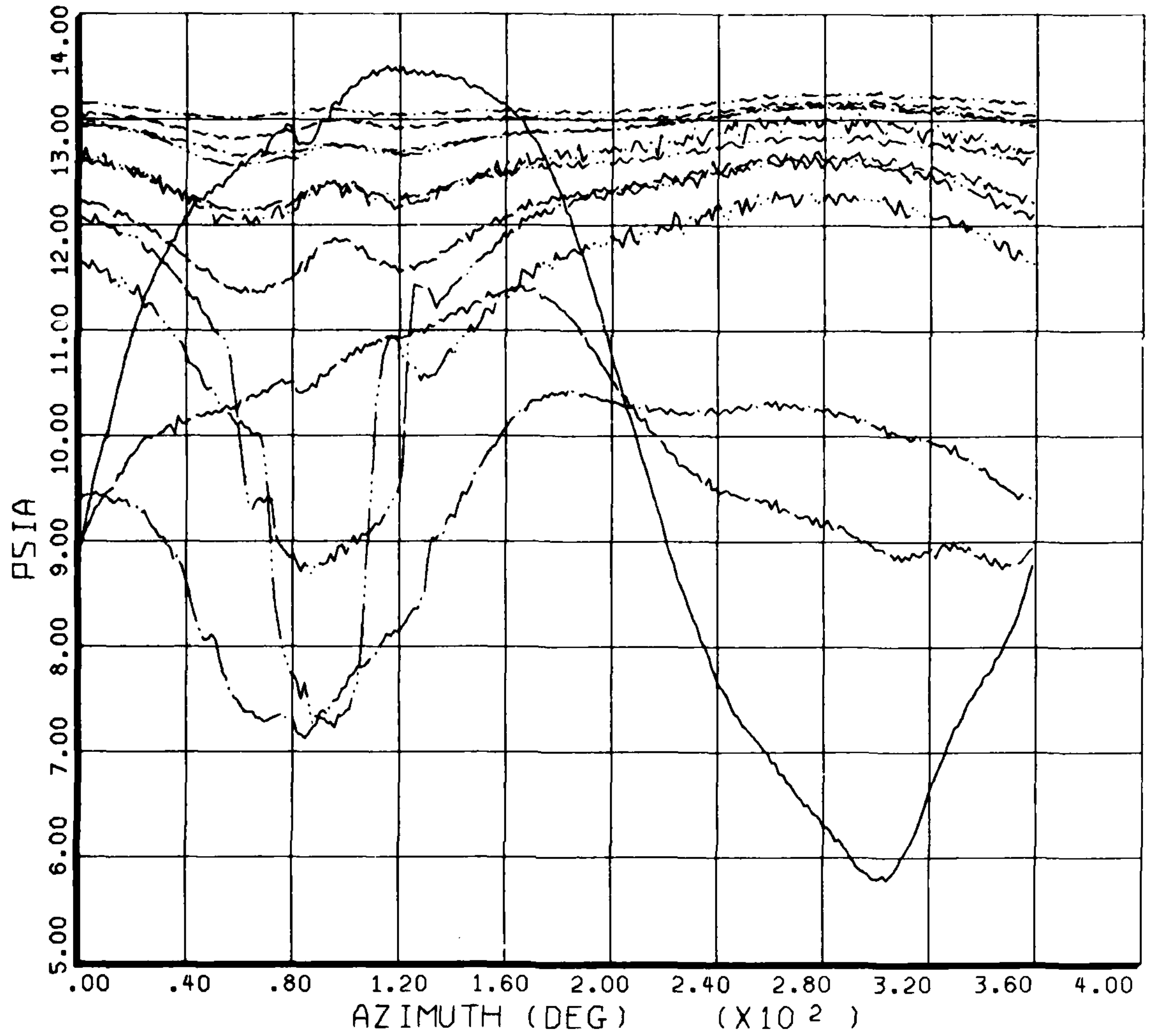
```
 NEW STEP
ANAL/FILTER 200 0 5/SCF1/KEEP SCF2/
 EXECUTING
```

A low-pass filter with an upper cut-off frequency of 200 Hz and 5 poles is selected. No graphic or printed output is generated. The user now regenerates the plot of data from the 86 percent span station using the filtered data from scratch file two.

```
 NEW STEP
DISP/SCF2 ALL ALL 4 TOP/MPLOT MRAZ/
 EXECUTING
```

The result of this command is the plot in Figure 1. Although some high-frequency behavior remains, the graph is much more easily read and the dash-dot pattern of the curves are easily discerned.

Scratch file two contains two functions of three independent variables. Upper and lower surface data are present as a function of azimuth, chord, and span. The user next proceeds to examine the functions in various "slices" or two-dimensional subsets of the three independent variables. First, a



LEVEL FLIGHT AT 129 KNOTS OLS DATA

CYCLE AVERAGE: BLADE ABSOLUTE PRESSURE

COUNTER	615	GROSS WT	8300	SHIP MODEL	AH-1G
.86	R/RADIUS	LONG CG	200.6	TOP SURFACE	

.01 X/CHORD
.03 X/CHORD
.08 X/CHORD
.20 X/CHORD
.25 X/CHORD
.35 X/CHORD
.40 X/CHORD
.45 X/CHORD
.50 X/CHORD
.55 X/CHORD
.60 X/CHORD
.70 X/CHORD

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/15/80

Figure 40. Cycle-averaged blade absolute pressure data with excessive high-frequency components.

contour plot of the upper surface data at 90 degrees azimuth is drawn.

```
 NEW STEP
DISP/SCF2 90 MRAZ ALL ALL TOP/
CONTOUR RECT,,,,,32/
 EXECUTING
```

Notice that the command is entered on two lines in this instance. Figure 2 is the result from this command. This contour plot is in the rectangular format as selected. Five commas follow the RECT keyword. These commas specify a sequence of four default values: IMPLIED, IMPLIED, AUTO, AUTO. The IMPLIED keywords indicate that the implied independent variable scales, normalized chord and normalized radius position, will be used. The AUTO keywords select automatic scaling of the contour levels. The "32" entry specifies that a maximum of 32 contour levels may be drawn. In fact, the automatic scaling selects 18 levels from 7.2 PSIA to 14.0 PSIA at intervals of 0.4 PSIA.

Next, the user requests a surface plot of the same data.

```
 NEW STEP
DISP/SCF2 90 MRAZ ALL ALL TOP/
SURFACE RECT,,,-10 -10 8/
 EXECUTING
```

Figure 3 is the result of this command. Notice in this plot that the intersection of the X, Y and Z axes has been moved from X=0, Y=0, Z=0, to the minimum X, Y, and Z values that occur on the plot (where Z is the dependent variable). This adjustment is made so that the X, Y, and Z axes will be visible on the plot frame for reference. The numeric entries "-10 -10 8", in the command specify the observers eye position relative to the displayed surface. Thus, the observers apparent eye position is 10 widths of the surface along the negative X axis, 10 widths of the surface along the negative Y axis, and 8 widths up on the Z axis. These distances are from the center of the depicted surface.

It is possible to view this surface from any angle. As an example, the user repositions the observer's eye at 2 widths on the X axis, 10 widths on the Y axis, and 4 widths up on the Z axis.

```
 NEW STEP
DISP/SCF2 90 MRAZ ALL ALL TOP/
SURFACE RECT,,, 2 10 4/
 EXECUTING
```

Figure 41 is the plot generated from this command.

Next, the user displays the top surface leading edge sensor data for all span stations and all azimuth positions.

```
 NEW STEP
DISP/SCF2 ALL 1 ALL TOP/SURF CYL MRAZ,,10 -10 7/
 EXECUTING
```

The surface plot generated by this command is displayed in Figure 4. Using the cylindrical format, the plot algorithm interprets the observer eye position numbers as positions on axes that are defined by azimuth directions. This command specified 10 widths in the zero degree direction, -10 widths in the 90-degree direction (i.e., +10 widths in the 270-degree direction), and 7 widths in the vertical direction.

Notice that MRAZ must be specified as the first independent variable for cylindrical format surface or contour plots.

The same data can be displayed in a cylindrical format contour plot.

```
 NEW STEP
DISP/SCF2 ALL 1 ALL TOP/CONT CYL MRAZ/
 EXECUTING
```

The resultant contour plot produced from this command is in Figure 5. The unspecified default values for the Disposition Substep are IMPL, AUTO, AUTO, 32. Notice that 32, rather than the original 16, is now the default maximum number of contour levels. The user also may display data from other chord stations. For example:

```
 NEW STEP
DIS/SCF2 ALL 6 ALL TOP/CONT CYL MRAZ/
 EXECUTING
```

Figure 42 is the result of this command showing the pressure measured at the 25 percent chord stations for all span stations. The same data can be drawn as a surface plot.

```
 NEW STEP
DISP/SCF2 ALL 6 ALL TOP/SURF CYL MRAZ,,7 10 6/
 EXECUTING
```

Figure 43 is the result of this command.

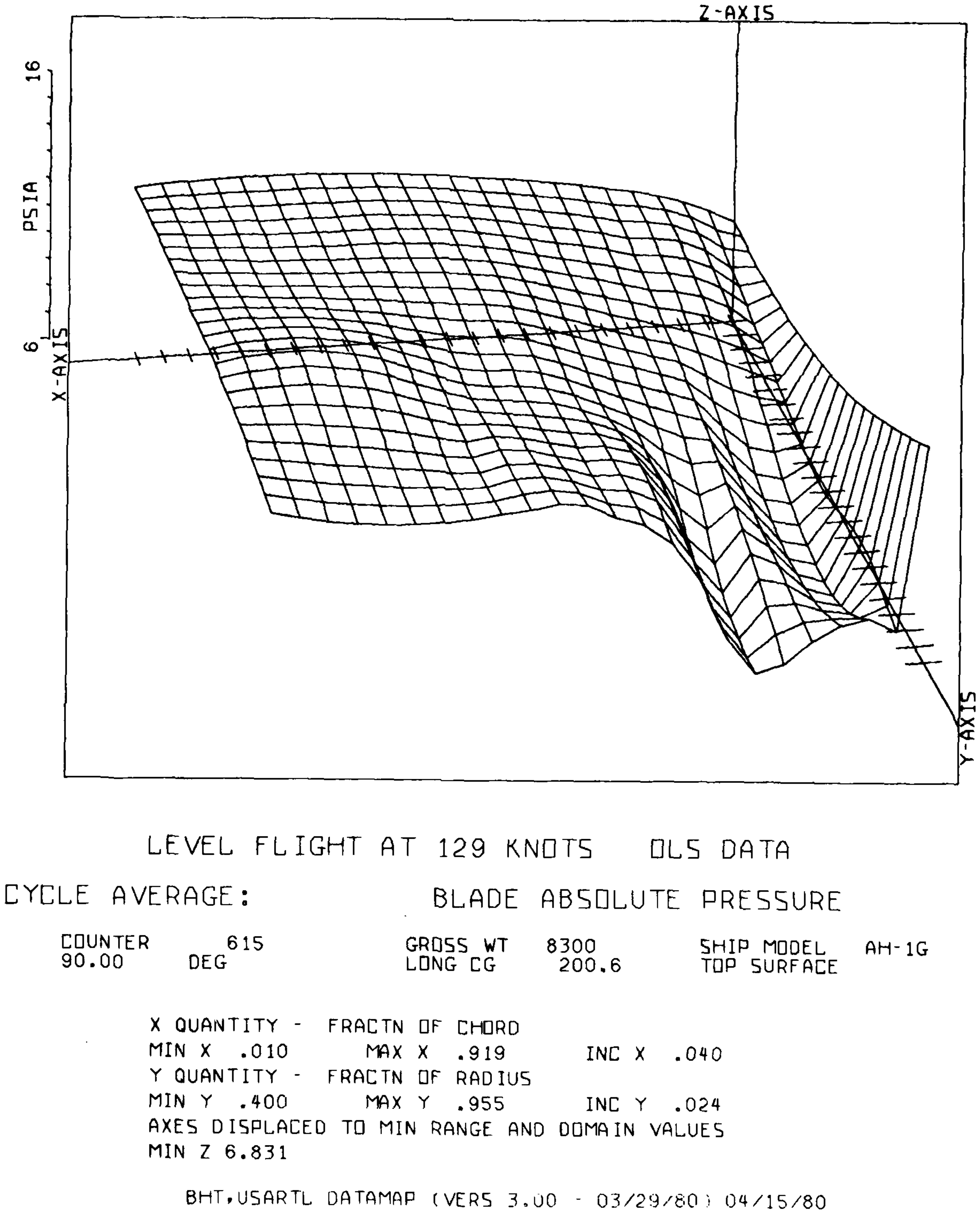
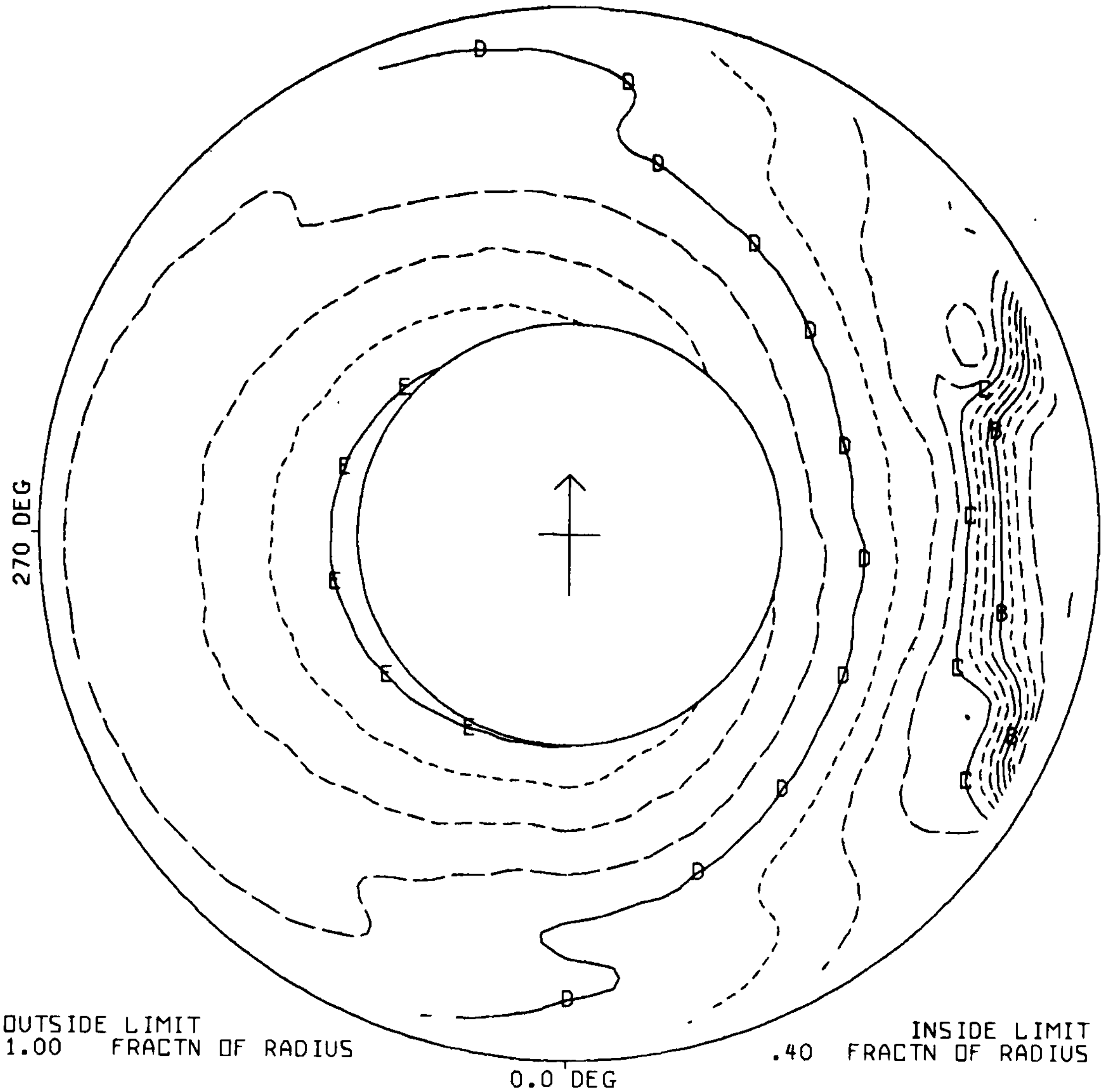


Figure 41. **Surface plot of blade absolute pressure on the top surface of the blade.**



LEVEL FLIGHT AT 129 KNOTS   OLS DATA

CYCLE AVERAGE: BLADE ABSOLUTE PRESSURE

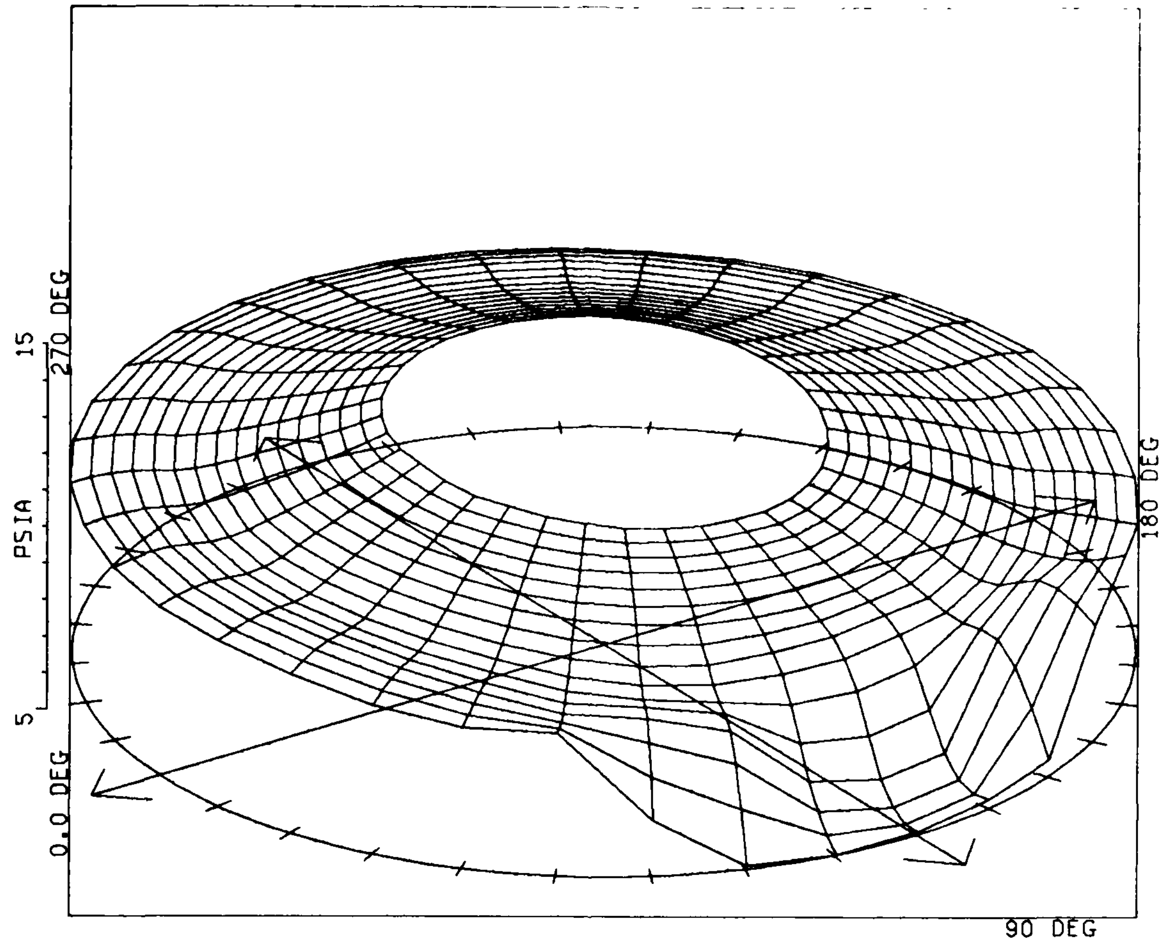
COUNTER	615	GROSS WT	8300	SHIP MODEL	AH-1G
.25	X/CHORD	LONG CG	200.6	TOP SURFACE	

--------------- CONTOUR LEVEL VALUES IN PSIA ---------------

A 6.8
B 8.4
C 10.0
D 11.6
E 13.2

BHT,USARTL DATAMAP (VERS 3.00 - 03/2?/80) 04/15/80

Figure 42. Contour plot of blade absolute pressure at 25 percent chord.



LEVEL FLIGHT AT 129 KNOTS   OLS DATA

CYCLE AVERAGE: BLADE ABSOLUTE PRESSURE

COUNTER	615	GROSS WT	8300	SHIP MODEL	AH-1G
.25	X/CHORD	LONG CG	200.6	TOP SURFACE	

ANGULAR INCREMENT 10 DEG
RADIAL QUANTITY   FRACTN OF RADIUS
MAX RADIUS   .955
RADIAL INCREMENT   .0370
ANGLE REFERENCE PLANE MOVED TO MIN 'Z' VALUE
MIN 'Z' = 6.801

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/15/80

Figure 43. Surface plot of blade absolute pressure at 25 percent chord.

5.7.2 Derivation of C_N and Comparison with C81 Data

This example represents a second or following session to further process the data that were examined in Paragraph 5.7.1. Using the permanent scratch files feature, the cycle-averaged pressure data in SCF2 from the previous session could be used as the starting point for this processing. However, the 1-percent chord top surface sensor at the 60-percent span station was not operating properly during the OLS experiment. The absence of data for this position can introduce considerable error in the Normal Force Coefficient, C_N, derivation at 60 percent span. Accordingly, for C_N processing, all of the 60 percent span item codes are omitted so that C_N will not be derived for this station. The item codes are omitted through the use of the Info File group S2PX instead of S2PP. S2PX does not include any of the item codes for the 60-percent span station. Thus, the cycle-averaging and filtering that was performed for the previous example must be repeated using the new Info File group, S2PX. First, the True Airspeed must be derived to establish the calibration curve factors for the new session since True Airspeed will be used in the computation of the Blade Static Pressure Coefficient, C_p. Reentry of these default values is required because command-specified defaults are not carried forward between sessions.

```
 NEW STEP
DERI/TAS,,,1.0208 3.33334/615 0 6/PLOT/
 EXECUTING
```

The plot in Figure 39 is recreated except that the grid is included. Next, the contents of SCF2 from Paragraph 5.7.1 are regenerated with the 60 percent span data omitted.

```
 NEW STEP
ANAL/AVER/GROUP S2PX,,,,615 0 6/KEEP SCF1/
 EXECUTING

 NEW STEP
ANAL/FILT 200 0 5/SCF1/KEEP SCF2/
 EXECUTING
```

Next, the C_p values are derived from the pressure data in scratch file two.

```
 NEW STEP
DERI/CP 264/SCF2/KEEP SCF3/
 EXECUTING
```

The blade radius, 264 inches, must be specified the first time that C_p is derived in a session.

The derived C_p values are stored in scratch file three and no graphic or printed output is produced by the step. The functions stored in SCF3 have a one-to-one correspondence in the independent variables to the functions in SCF2. That is, there are top and bottom surface functions of azimuth, chord, and span. The user proceeds to examine this data at 90 degrees azimuth. First, the user enters a descriptive comment for the flight condition.

```
 NEW STEP
COMM/LEVEL FLIGHT AT 129 KNOTS  OLS DATA/
```

```
 NEW STEP
DISP/SCF3 90 MRAZ ALL ALL TOP/MPLOT/
 EXECUTING
```

The resultant plot is displayed in Figure 6. If the user wishes to view the C_p values for both the top and bottom surface in one plot, the DPLOT output option may be used.

```
 NEW STEP
DISP/SCF3 90 MRAZ ALL ALL TOP/DPLOT/
 EXECUTING
```

Figure 7 shows the plot that is produced from this command.

Next, the user derives the Normal Force Coefficient, C_N, from the C_p values in scratch file three and stores the results in scratch file two.

```
 NEW STEP
DERI/CN/SCF3/KEEP SCF2/
 EXECUTING
```

The previous contents of scratch file two, blade absolute pressure data, are overwritten during execution of this command and destroyed. The new content of scratch file two is a single function, C_N, of the independent variables azimuth and span. The reduction in the number of independent variables is accomplished by the C_N integration around the blade at each span station. Thus, the entire contents of scratch file two can be displayed in a single plot.

```
 NEW STEP
DISP/SCF2/CONT CYL MRAZ/
 EXECUTING
```

Figure 44 is the resultant plot from this command. The same data can be displayed in a surface plot.

```
 NEW STEP
DISP/SCF2/SURF CYL MRAZ,,10 10 10/
 EXECUTING
```

Figure 9 is the plot drawn from this command.

Part of the objective for this example is to compare OLS and C81 data. Accordingly, access is made to a Master File partition that contains a C81 simulation of the OLS flight condition represented in counter 615. The name for this partition is "PARTNAME".

```
 NEW STEP
UTILITY/PARTITION/PARTNAME SECOND/
 'PARTNAME' ACCESSED AS SECOND PARTITION
```

Thus, the partition, "PARTNAME", is accessed through the second partition access slot, and this fact is confirmed by the return message from the computer. Access to the original partition data that contains the OLS data is retained as confirmed by a menu listing of counters.

```
 NEW STEP
MENU/DATA/
```

The counters 610 through 615, 360083, 360084, and 360085 are listed. From C81 run output, the user knows that counter 360084 represents 129 knots. The user displays the C81 simulated C_N values from this counter. First, the user enters an appropriate comment for the C81 data.

```
 NEW STEP
COMM/LEVEL FLIGHT AT 129 KNOTS  C81 ANALYSIS/
```

```
 NEW STEP
ANAL/AVER/GROUP CNR1,,,,360084 .1 1/
CONT CYL MRAZ/
 EXECUTING
```

Figure 45 is the result of this command. Notice the unusual behavior of the depicted function for small radii. The C81 generated values for C_N are a constant zero for the first four span stations, which results in unusual values from the interpolation method that is used for three-dimensional plots.

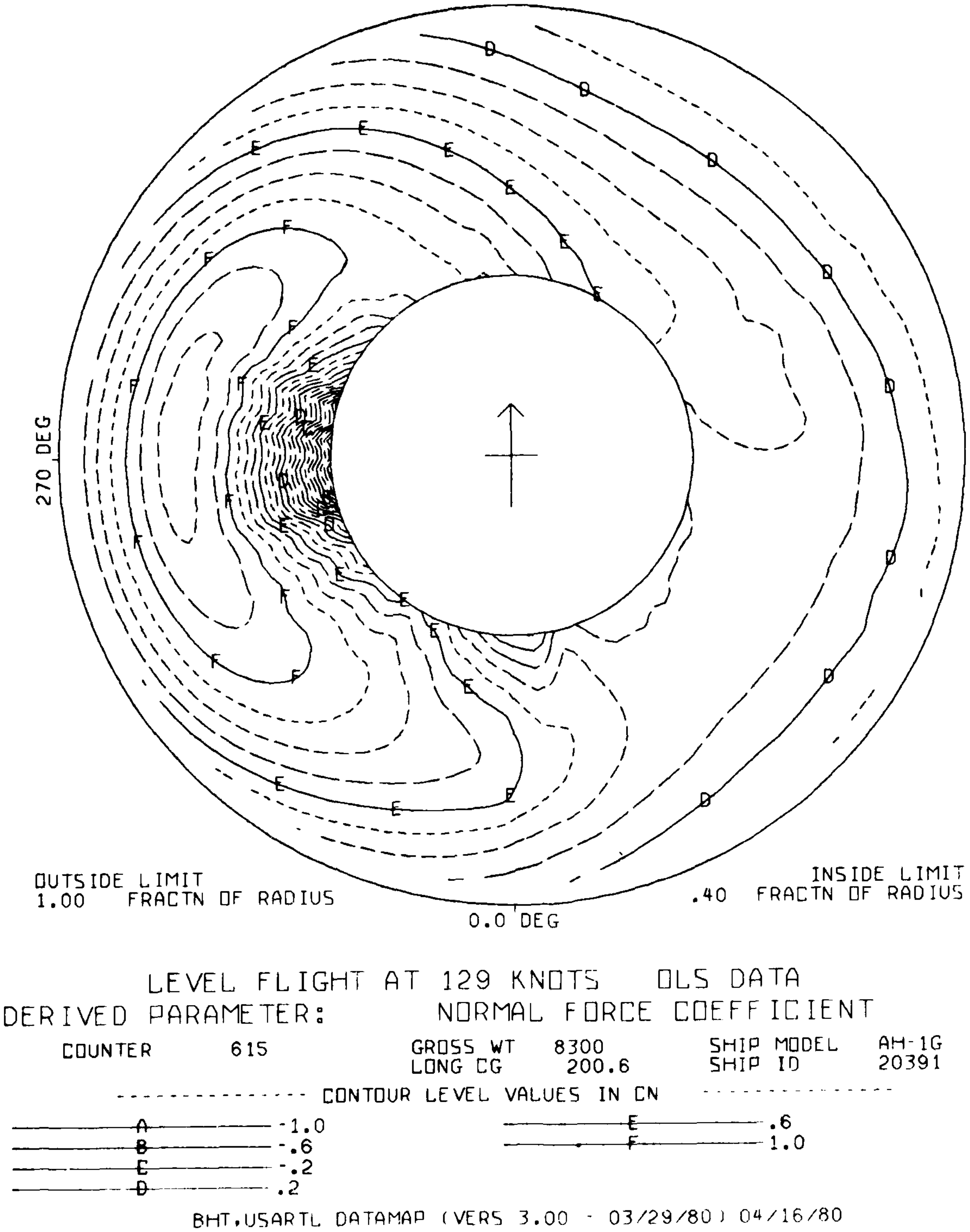


Figure 44. Contour plot of blade normal force coefficient.

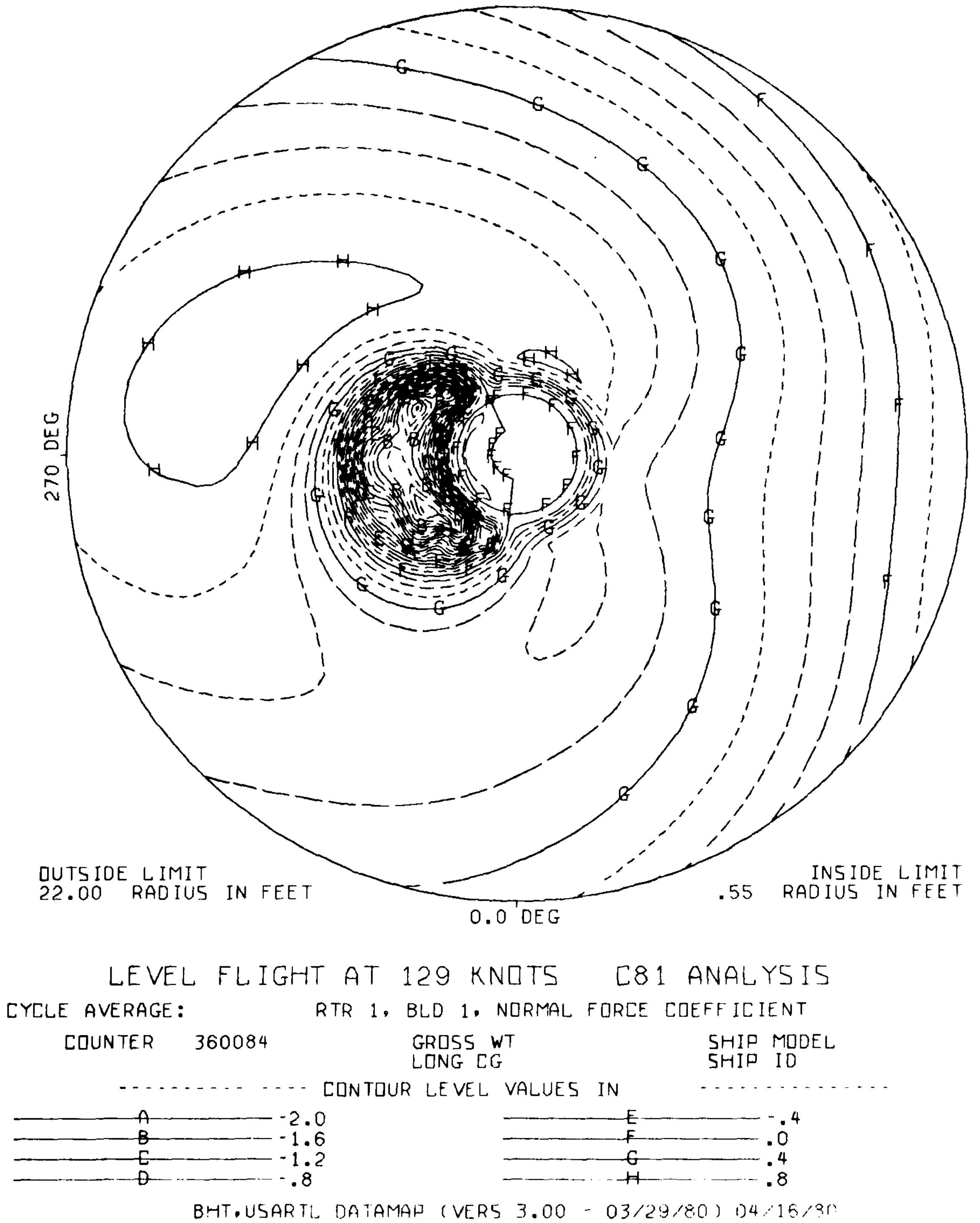


Figure 45. Contour plot of C81-generated blade normal force coefficient.

Since the data from these low-span stations (i.e., stations on the hub) are not useful, the user masks the corresponding item codes.

```
 NEW STEP
UTILITY/MASK/B662/

 NEW STEP
UTIL/MASK/B661/

 NEW STEP
UTIL/MASK/B660/

 NEW STEP
UTIL/MASK/B659/
```

Notice that a separate mask command is required for each item code. The C81 generated C_N data are now replotted.

```
 NEW STEP
ANAL/AVER/GROUP CNR1,,,,360084 .1 1/
CONT CYL MRAZ/
 EXECUTING
```

Figure 46 is the plot generated from this command. Notice that when data are missing from a section of the independent variable domain and extrapolation is required to obtain values for the section, the section is simply left blank in contour and surface plot output. The same data from Figure 46 can be drawn as a surface plot.

```
 NEW STEP
ANAL/AVER/GROUP CNR1,,,,360084 .1 1/
SURF CYL MRAZ,,10 10 10/
 EXECUTING
```

This command generates the plot in Figure 11. The last two generated plots, Figures 11 and 46, are useful for examination of the C81 data itself. However, for comparison with the OLS data, it is preferrable to limit the domain of the C81 data to the domain of the OLS data. The first OLS span position is at 40 percent span. Thus, four more item codes are masked so that the first active item code in the radial direction is at 39 percent span.

```
 NEW STEP
UTIL/MASK/B658/

 NEW STEP
UTIL/MASK/B657/
```

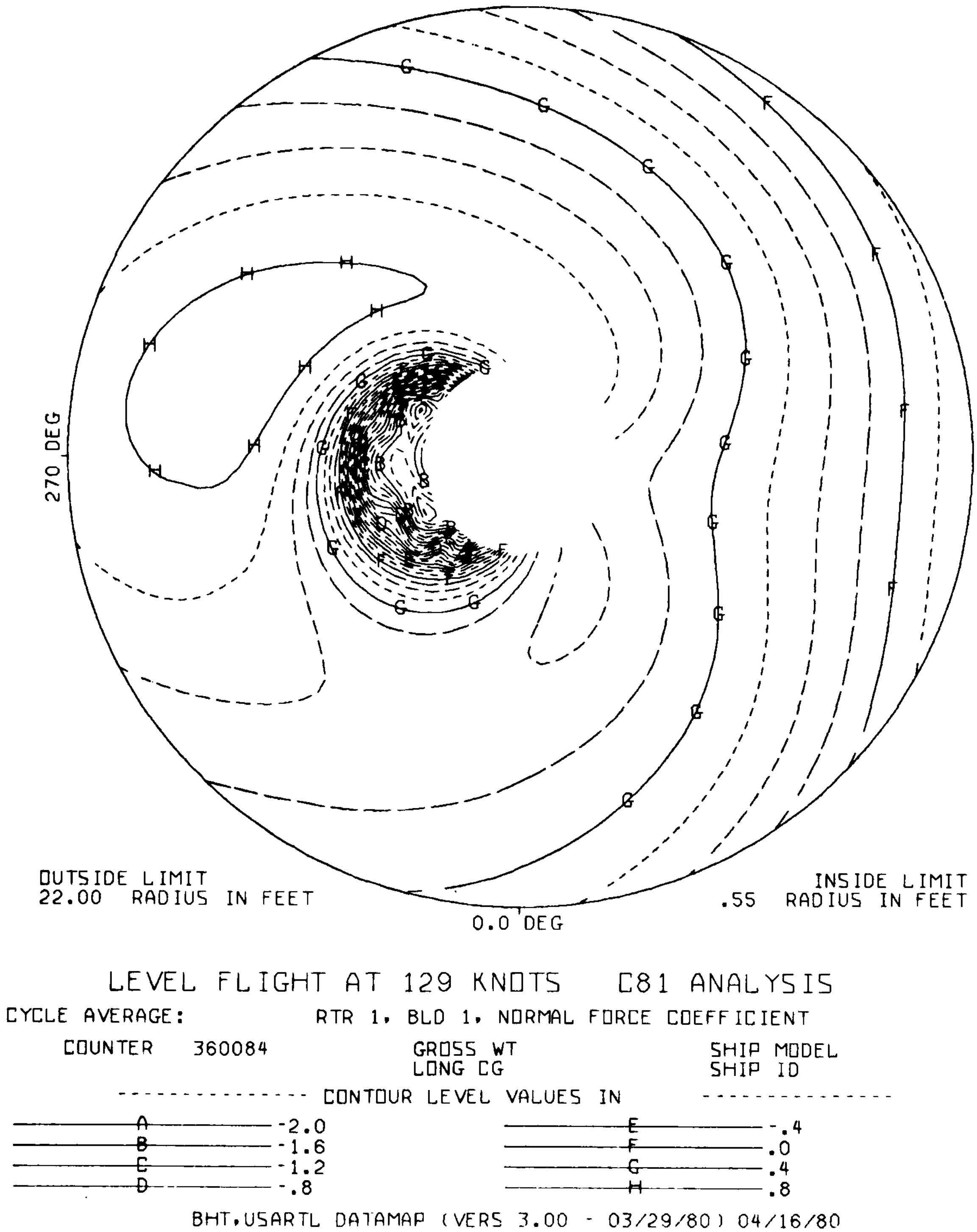


Figure 46. Contour plot of C81-generated blade normal force coefficient with some item codes masked.

```
 NEW STEP
UTIL/MASK/B656/

 NEW STEP
UTIL/MASK/B655/
```

Now a contour plot is to be generated for direct comparison with OLS data. For this purpose, it is desirable that the contour level labels correspond for the two plots. Thus, direct specification of the level scaling will be used for both a C81 data plot and an OLS data plot.

```
 NEW PLOT
ANAL/AVER/GROUP CNR1,,,,360084 .1 1/
CONT CYL MRAZ,, .1, -1.2, 32/
 EXECUTING
```

Figure 10 is generated from this command. In the Disposition Substep, .1 is the interval between contour levels and -1.2 is the first level that the computer will attempt to draw. Since the C_N values for the depicted function do not extend to that low a value, no contours are drawn for that level. However, a sample contour line for that level is drawn in the annotation area below the plot. The corresponding OLS data can be drawn using the same scale. First, the comment must be changed to indicate that OLS data are displayed.

```
 NEW STEP
COMM/LEVEL FLIGHT AT 129 KNOTS  OLS DATA/

 NEW STEP
DISP/SCF2/CONT CYL MRAZ,, .1, -1.2, 32/
 EXECUTING
```

Figure 8 is generated from this command. Notice that each labeled contour line represents the same level as in Figure 10.

Next, the user wishes a direct comparison of C_N values depicting OLS and C81 data on the same plot. This comparison can be done on an X-Y plot. For comparisons between data bases, it is desirable to use the LPLOT output option so that sufficient annotation can be included to properly identify each curve on the plot. The user adjusts the comment to reflect the 40-percent span station and then enters a command to draw the OLS C_N values for the 40-percent span station.

```
 NEW STEP
COMM/OLS DATA FOR 129 KNOTS 40 PERCENT SPAN/
```

```
 NEW STEP
DISP/SCF2 ALL ALL 1/LPLOT MRAZ/
 EXECUTING
```

Figure 47 is the initial result of this command. Now the C81 data for the 39-percent span station can be added to the plot. First, the comment must be adjusted to describe the C81 data.

```
 NEW STEP
COMM/C81 ANALYSIS FOR 129 KNOTS 39 PERCENT SPAN/

 NEW STEP
ANAL/AVER/GROUP CNR1 TOP 9 ALL 360084 .1 1/
APLOT MRAZ/
 EXECUTING
```

Figure 12 is the final result from this command showing both curves on the same plot.

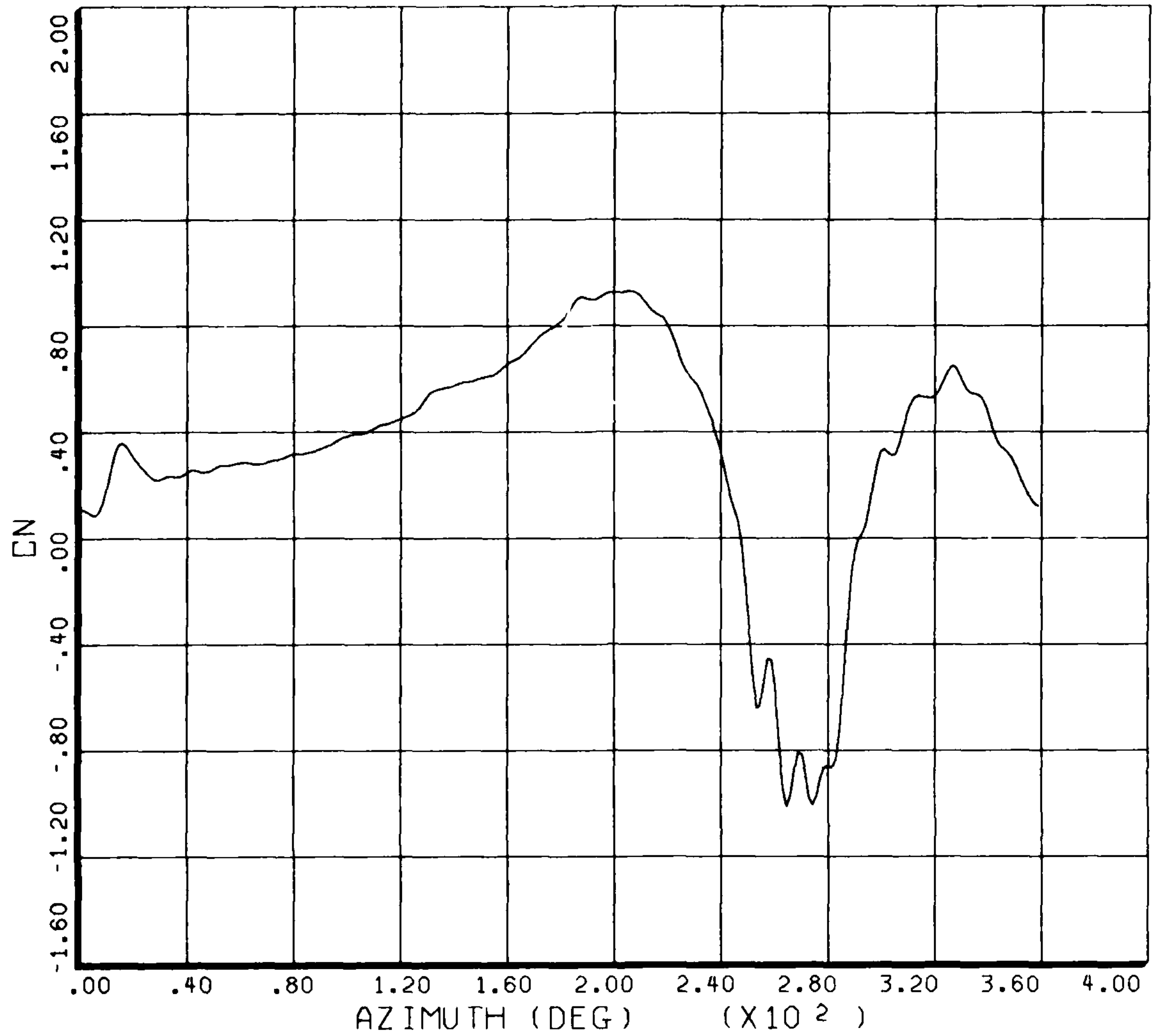
5.7.3 Horsepower Derivation and Display Versus True Airspeed

Consider the same counters (i.e., flight conditions) that were available for the example in Paragraph 5.7.2, but assume a new interactive session using the Processing Program. Access is initially established to the same partitions that were addressed from the example in Paragraph 5.7.2. The OLS counters, 610 through 615, represent airspeeds of 72 knots through 143 knots. The C81 counters, 360083 through 360085, are simulations of airspeeds of 116 knots through 143 knots. Mast horsepower is available directly for the C81 counters as item code A133 and may be derived from mast torque, which is available for the OLS data. Assuming that the user is interested in a comparison of mast horsepower, he may derive and store horsepower data for each counter. The airspeed calibration constants should first be entered through the device of deriving True Airspeed for a counter.

```
 NEW STEP
DERI/TAS,,,1.0208 3.33334/615 0 6/PLOT/
 EXECUTING
```

A plot similar to Figure 39 is generated except that a grid is drawn. Next, mast horsepower is derived for each of the OLS counters and each resultant time history is stored on scratch file one. Multiple commands are required to perform these derivations.

```
 NEW STEP
DERIVE/MSHP/610 0 6/KEEP SCF1/
 EXECUTING
```



______________	COUNTER	615	GROSS WT	8300	SHIP MODEL	AH-1G
			LONG CG	200.6	SHIP ID	20391

OLS DATA FOR 129 KNOTS 40 PERCENT SPAN
DERIVED PARAMETER: NORMAL FORCE COEFFICIENT

BHT.USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/16/80

Figure 47. LPLOT with one curve showing OLS C_n data.

```
 NEW STEP
DERI/MSHP/611 0 6/ADD SCF1/
 EXECUTING

 NEW STEP
DERI/MSHP/612 0 6/ADD SCF1/
 EXECUTING

 NEW STEP
DERI/MSHP/613 0 6/ADD SCF1/
 EXECUTING

 NEW STEP
DERI/MSHP/614 0 6/ADD SCF1/
 EXECUTING

 NEW STEP
DERI/MSHP/615 0 6/ADD SCF1/
 EXECUTING
```

The preceding commands load the derived main rotor shaft horsepower for each of the counters onto scratch file one. Each of the counters is loaded as a separate column element in the file. Thus, the time history from counter 611 is the second column element in the file. Notice that for the first command, the keyword "KEEP" is used in the Disposition Substep. This entry selection specifes that the data that are stored overwrite the previous contents of the file, which are destroyed. The "ADD" keyword used in the subsequent commands specified that output is written to the scratch file in addition to the data that are already stored. The output from each command with an ADD Disposition Substep is stored as one column element in the scratch file.

All of the mast horsepower time histories that are stored on scratch file one may be displayed on a multiple curve plot. First, a descriptive comment is added.

```
 NEW STEP
COMM/OLS MEASURED DATA  HORSEPOWER SWEEP/

 NEW STEP
DISP/SCF1/MPLOT/
 EXECUTING
```

Figure 48 is produced from this command. Notice that each time history has a different start time in this plot. This staggering of start times reflects the fact that mast horsepower was derived in complete rotor cycles and these cycles

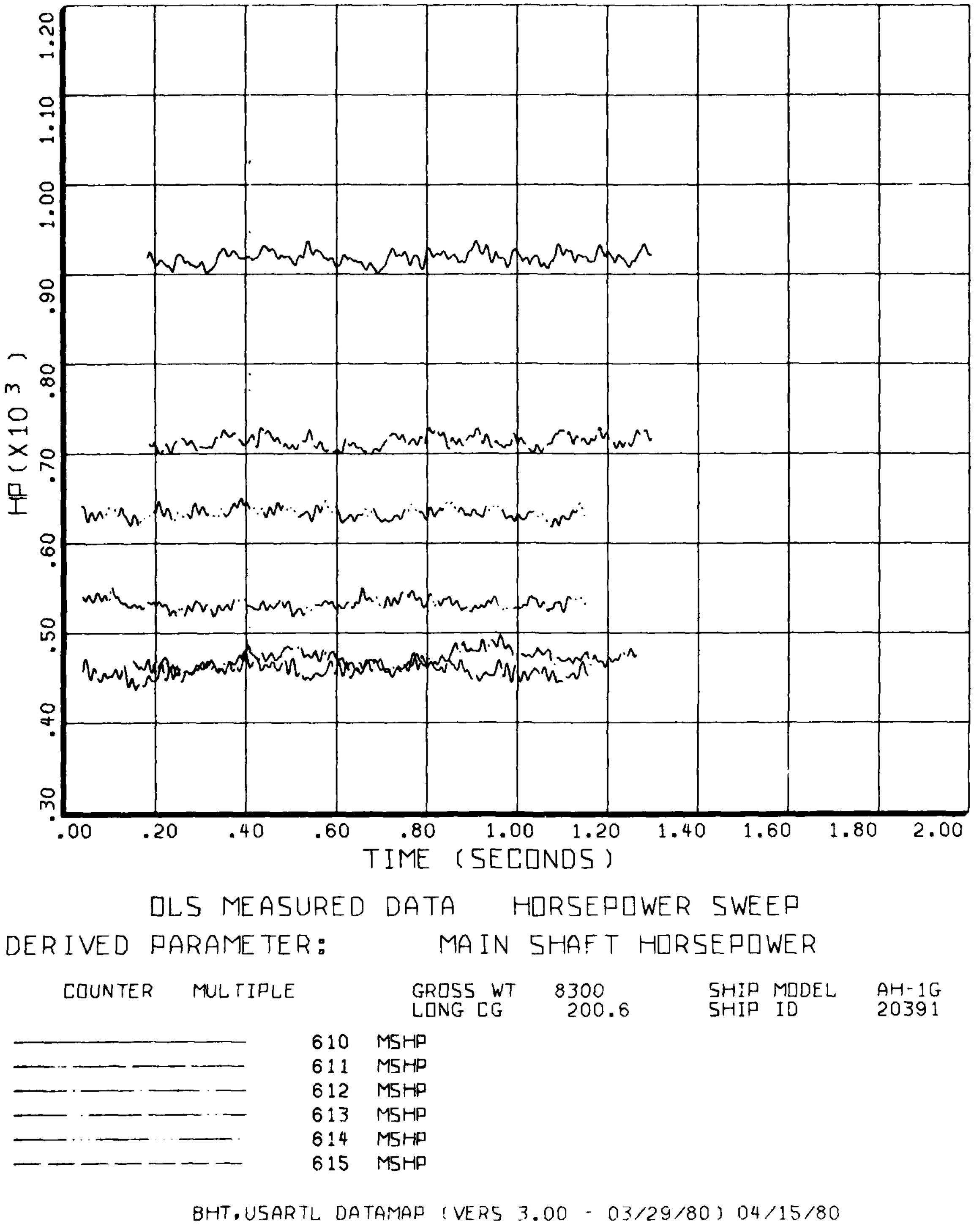


Figure 48. Mast horsepower records for multiple counters.

have different start times for the different counters. Also notice that there are large higher harmonic components in the time histories that make the plot difficult to read. Since only the mean value and long-term variation of the data are of interest, the time histories can be filtered and re-plotted.

```
 NEW STEP
ANAL/FILT 2.5 0 7/SCF1/KEEP SCF2/
 EXECUTING

 NEW STEP
DISP/SCF2/MPLOT/
 EXECUTING
```

Figure 49 is the result of these two commands. The curves can be more easily differentiated on this plot and the horse-power levels appear steady. These levels can now be displayed as a function of airspeed. A mean value is first computed for each curve and the LPLOT output option is used so that C81 data can be added later.

```
 NEW STEP
ANAL/STATISTICS MEAN/INDIVIDUAL SCF2/LPLOT TAS/
 EXECUTING
```

The plot generated by this command is displayed in Figure 50. The user must specify "INDIVIDUAL" as the ensemble averaging option so that all of the scratch file column elements are not averaged together. To add a corresponding curve for the C81 analysis, the user must load the C81 main shaft horse-power data on a scratch file for the three flight conditions available.

```
 NEW STEP
DISPLAY/A133 360083 0 1/KEEP SCF3/
 EXECUTING

 NEW STEP
DISP/A133 360084 0 1/ADD SCF3/
 EXECUTING

 NEW STEP
DISP/A133 360085 0 1/ADD SCF3/
 EXECUTING
```

Notice that the DISPLAY command can be used to move data to a scratch file without any analysis or derivation processing. After execution of the above commands, scratch file three contains main shaft horsepower for three different simulated

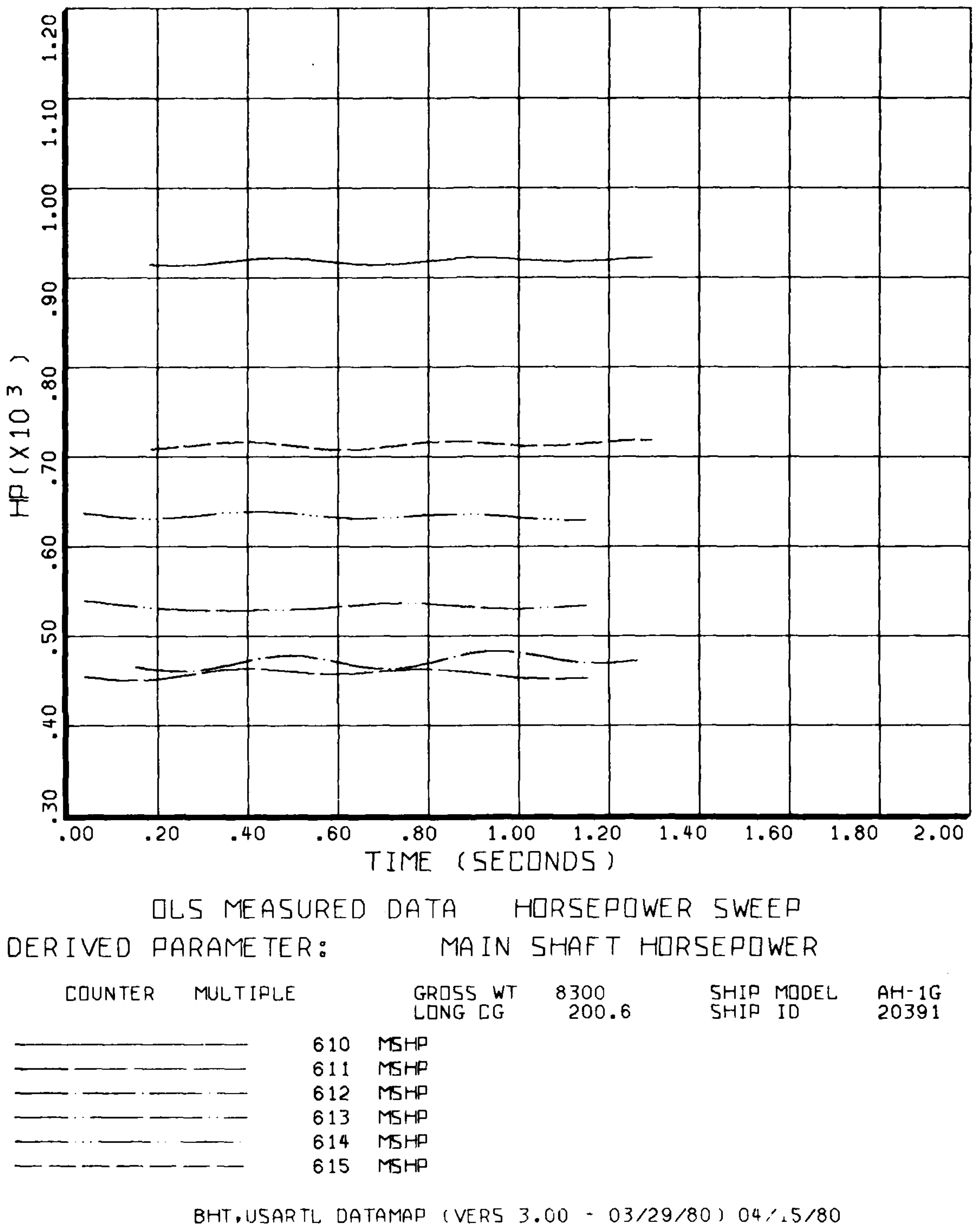


Figure 49. Filtered mast horsepower records for multiple counters.

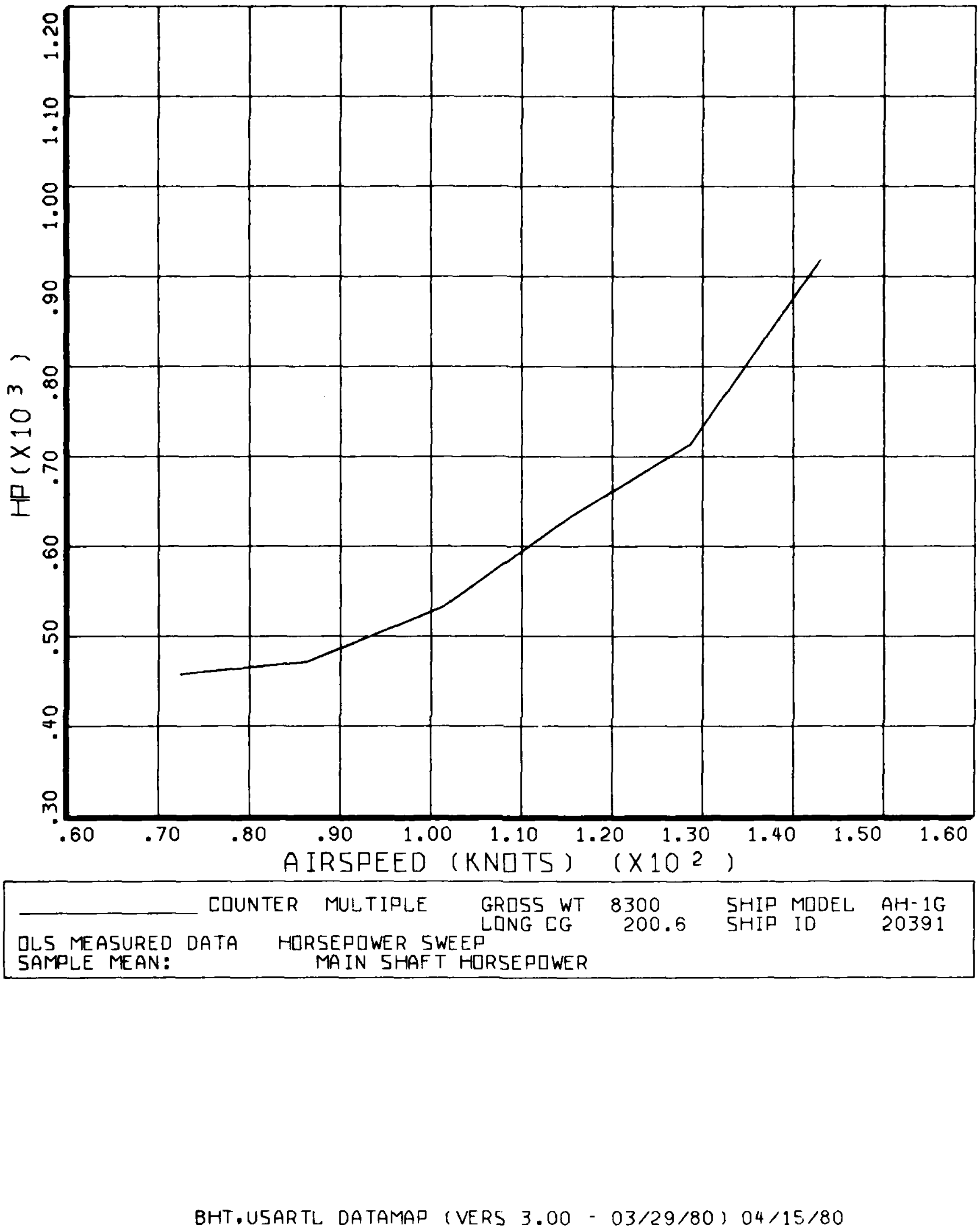


Figure 50. Horsepower versus airspeed for OLS data.

flight conditions. The mean horsepower values for these flight conditions can also be plotted versus airspeed. First, an appropriate comment should be entered for the C81 data.

```
 NEW STEP
COMM/C81 ANALYSIS  HORSEPOWER SWEEP/
 EXECUTING

 NEW STEP
ANAL/STAT MEAN/IND SCF3/APLOT TAS/
 EXECUTING
```

Figure 51 shows the plot after the C81 curve and annotation have been added.

5.7.4 Building, Modifying, and Executing Command Sequences

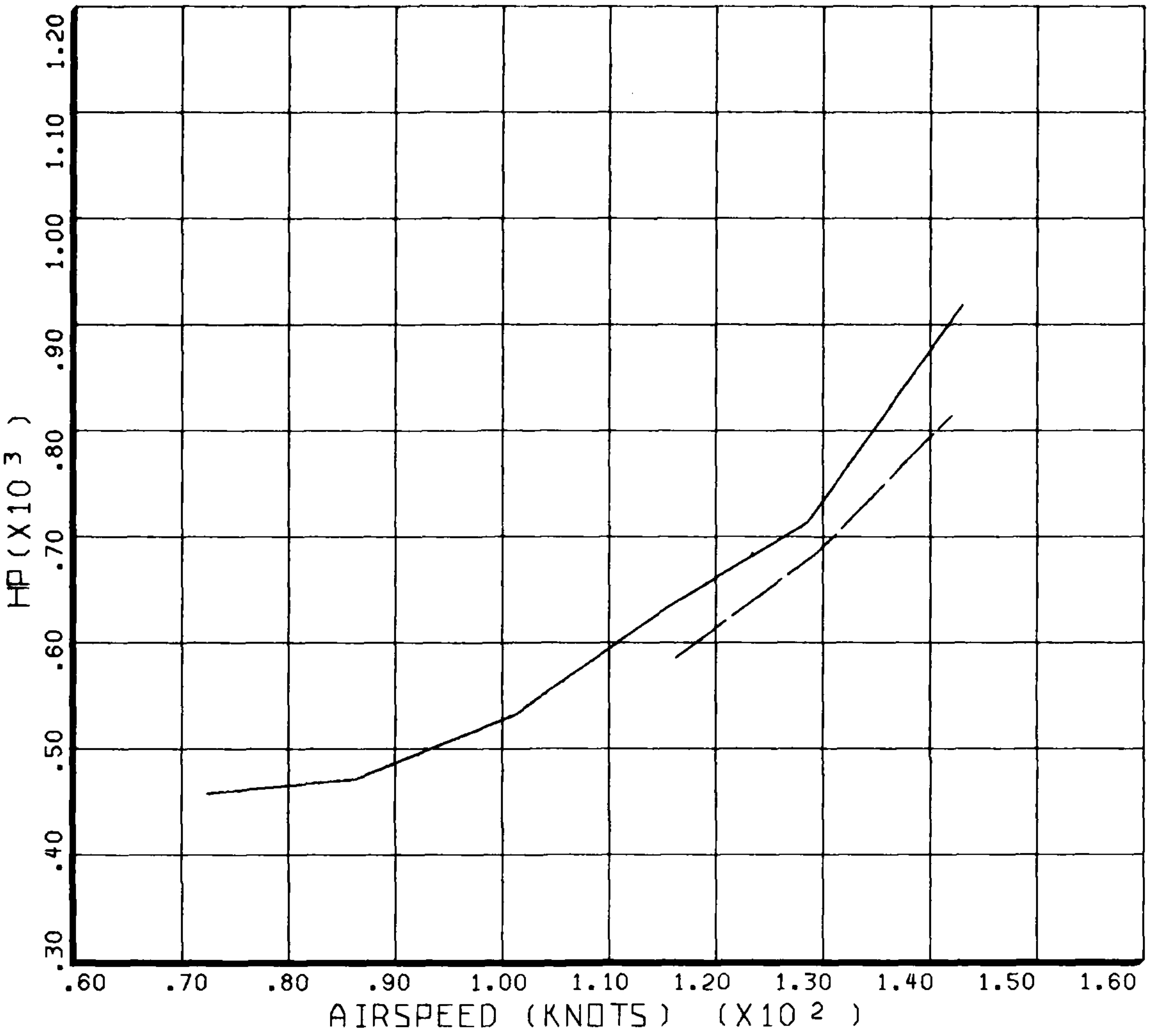
The command sequence execution capability can be of significant value for Processing Program operation in batch mode, particularly when the parameter passing capability is used. The normal sequence of events for use of command sequences is to create a command sequence in the interactive graphics mode, modify the command sequence to accept parameters, and then execute the command sequence for several different counters in batch mode. This paragraph illustrates the method.

Assume that the user wishes to do processing similar to that described in Paragraph 5.7.2, but that less comprehensive plot output is desired. In particular, assume that the user wishes contour plots of C_N for the six different airspeeds of counters 610 through 615. First, the user executes the Processing Program in the interactive graphics mode. The initial command entered is

```
 NEW STEP
MENU/EDIT/
```

The computer lists the names of the command sequence blocks (i.e., edit blocks) that are in the command sequence storage file.

```
COMMAND SEQUENCE FILE
MAX NUMBER OF COMMAND SEQUENCE
BLOCK NAMES ALLOWED IS 6
EXISTING BLOCK NAMES ARE:
     APL1
     BBBB
     CCCC
     DDDD
     EEEE
     FFFF
```



______________ COUNTER MULTIPLE	GROSS WT	8300	SHIP MODEL	AH-1G
OLS MEASURED DATA HORSEPOWER SWEEP	LONG CG	200.6	SHIP ID	20391
SAMPLE MEAN: MAIN SHAFT HORSEPOWER				
____ ____ ____ COUNTER MULTIPLE	GROSS WT		SHIP MODEL	
C81 ANALYSIS HORSEPOWER SWEEP	LONG CG		SHIP ID	
SAMPLE MEAN: ROTOR 1, HORSEPOWER				

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/15/80

Figure 51. Comparison of horsepower for OLS and C81 versus airspeed.

This menu listing indicates that there is space for six command sequence blocks and six blocks are already present. If a new command sequence is to be created, then one of the existing blocks must be deleted first. For example, if the block "BBBB" is no longer necessary, this block can be deleted.

```
 NEW STEP
EDIT/DELETE BBBB/
 COMMAND SEQUENCE 'BBBB' DELETED
```

Now there is a free slot for storage of a command sequence. The user elects to generate this block in the "BUILD" mode so that each command is executed interactively after entry and then stored at the users option using the "SAVE" command.

```
 NEW STEP
BUILD/CNPL/
 ENTERING 'BUILD/SAVE' MODE WITH COMMAND SEQUENCE NAME 'CNPL'
```

Now the user proceeds to execute the required steps and save the steps on the command sequence block. The processing flow is similar to the flow for Paragraph 5.7.2 except that no plots of intermediate results are generated and the derived C_N values are stored in scratch file one. The true airspeed derivation is performed first and not saved because the airspeed calibration factors need be entered only once in a Processing Program run, assuming that these factors are correct for the entire airspeed range that is being examined.

```
 NEW STEP
DERI/TAS,,, 1.0208 3.33334/615 0 6/PLOT/
 EXECUTING
```

A plot similar to Figure 39 is again generated. Now the commands to be saved are entered.

```
 NEW STEP
ANAL/AVER/GROUP S2PX,,,,615 0 6/KEEP SCF1/
 EXECUTING

 NEW STEP
SAVE/

 NEW STEP
ANAL/FILT 200 0 5/SCF1/KEEP SCF2/
 EXECUTING

 NEW STEP
SAVE/
```

```
 NEW STEP
DERI/CP 264/SCF2/KEEP SCF3/
 EXECUTING

 NEW STEP
SAVE/

 NEW STEP
DERI/CN/SCF3/KEEP SCF1/
 EXECUTING

 NEW STEP
SAVE/

 NEW STEP
COMM/LEVEL FLIGHT AT 129 KNOTS  OLS DATA/

 NEW STEP
SAVE/

 NEW STEP
DISP/SCF2/CONT CYL MRAZ,, .1 -1.2 32/
 EXECUTING
 ***SELECTED OUTPUT MODE CONFLICTS IN
 ***DIMENSION WITH THE INPUT/PROCESS
 **DISPOSITION SETUP ERROR 475
```

13 F The error encountered for the last step illustrates the advantages of the "BUILD/SAVE" mode. The error was detected during processing of the step rather than during interpretation of the command. Thus, this error would not have been detected while using the "EDIT/NEW" method of command sequence generation. The error detected was that scratch file two contains functions of three independent variables and the default entries in the Input Substep specify that all of these independent variables should be passed through for output. The contour plot output option will only accommodate two independent variables. The user meant to specify SCF1 for input, which contains C_N as a function of two independent variables (i.e., azimuth and radius). The user does not enter a SAVE step so that the erroneous command is not saved on the command sequence file. Instead, the user reenters the command with the correct scratch file input.

```
 NEW STEP
DISP/SCF1/CONT CYL MRAZ,, .1 -1.2 32/
 EXECUTING
```

Figure 8 is recreated on the Tektronix screen and, if the user decides that the plot is correct, the user can save the step and close out the command sequence block

```
 NEW STEP
SAVE/

 NEW STEP
NOEDIT/
```

Now, for application of the command sequence "CNPL" to counters other than 615, the user must enter the "EDIT/CHANGE" mode to condition the sequence to recieve parameters.

```
  NEW STEP
EDIT/CHANGE CNPL/
 EDIT CHANGE MODE SUB-COMMANDS:
 $A,LINE1                ADD LINE(S)
 $C,LINE1,LINE2          CHNG/DELETE LINE(S)
 $L,LINE1,LINE2          LIST LINE(S)
 $N                      RENUMBER
 $E                      END
 $?                      HELP
 EDIT CHANGE MODE.
 ENTER ONE OF
 #A,$C,$L,$N,$E,$?
```

The first change mode subcommand entered by the user requests a listing of the command sequence block.

```
$L
**********CNPL  LIST**********                                  13
******08.18.49  04/18/80******                                  B

   1 ANAL/AVER/GROU/S2PX,BOTH,ALL ,ALL , 615,0,6, -.01/KEEP
   2 SCF1,NONE/
   3 ANAL/FILT, 200,0, 5/SCF1,ALL ,ALL ,ALL ,BOTH/KEEP,SCF2,
   4 NONE/
   5 DERI/CP , 264,CALC,CALC/SCF2,ALL ,ALL ,ALL ,BOTH/KEEP,
   6 SCF3,NONE/
   7 DERI/CN /SCF3,ALL ,ALL ,ALL /KEEP,SCF1,NONE/
   8 COMM/LEVEL FLIGHT AT 129 KNOTS  OLS DATA/
   9 DISP/SCF1,ALL ,ALL ,ALL ,BOTH/CONT,CYLI,MRAZ,IMPL, .1,
  10 -1.2, 32/
  11 NOED/
  EDIT CHANGE MODE.
  ENTER ONE OF
  $A,$C,$L,$N,$E,$?
```

Notice that, as an exception to our convention, the asterisks typed by the computer begin in column one. The user wishes to parameterize four entries in this list: the counter, the airspeed, and the two scale factors for the contour plot. These entries occur on lines 1, 8, 9 and 10. Each of these lines must be completely retyped in order to change the entries. The "$C" subcommand is used to change these lines.

```
$C 1
ANAL/AVER/GROU/S2PX,BOTH,ALL,ALL, %1, 0, 6, -.01/KEEP
$C 8 10
COMM/LEVEL FLIGHT AT %2 KNOTS  OLS DATA/
DISP/SCF1,ALL,ALL,ALL,BOTH/CONT,CYLI,MRAZ,IMPL,%3,
 %4,32/
```

Then the "$N" command is used to renumber the lines and the $L command is entered to list the complete command sequence. The sequence must be renumbered to add line numbers to the changed lines.

```
$N
 EDIT CHANGE MODE
 ENTER ONE OF
 $A,$C,$L,$N,$E,$?

$L
**********CNPL  LIST**********
******08.21.21  04/18/80******

  1 ANAL/AVER/GROU,S2PX,BOTH,ALL,ALL,%1,0,6, -.01/KEEP
  2 SCF1,NONE/
  3 ANAL/FILT,200,0, 5/SCF1,ALL ,ALL ,ALL ,BOTH/KEEP,SCF2,
  4 NONE/
  5 DERI/CP ,264,CALC,CALC/SCF2,ALL ,ALL ,ALL ,BOTH/KEEP,
  6 SCF3,NONE/
  7 DERI/CN /SCF3,ALL ,ALL ,ALL /KEEP,SCF1,NONE/
  8 COMM/LEVEL FLIGHT AT %2 KNOTS  OLS DATA/
  9 DISP/SCF1,ALL ,ALL ,ALL ,BOTH/CONT,CYLI,MRAZ,IMPL,%3,
 10 %4,32/
 11 NOED/
 EDIT CHANGE MODE.
 ENTER ONE OF
 $A,$C,$L,$N,$E,$?
```

The command sequence block appears correct as listed so the user exits from the change mode.

```
$E

 ENTER STORE OR KILL

STORE

 NEW STEP
```

The user is prompted to enter one of the two keywords "STORE" or "KILL". "KILL" means that any modifications made during the "EDIT/CHANGE" session are ignored and the original command sequence is left unchanged. "STORE" means that the command sequence as currently modified is stored on the command sequence file. The user selects "STORE" and the program

returns to the normal command entry mode as indicated by the "NEW STEP" message from the computer.

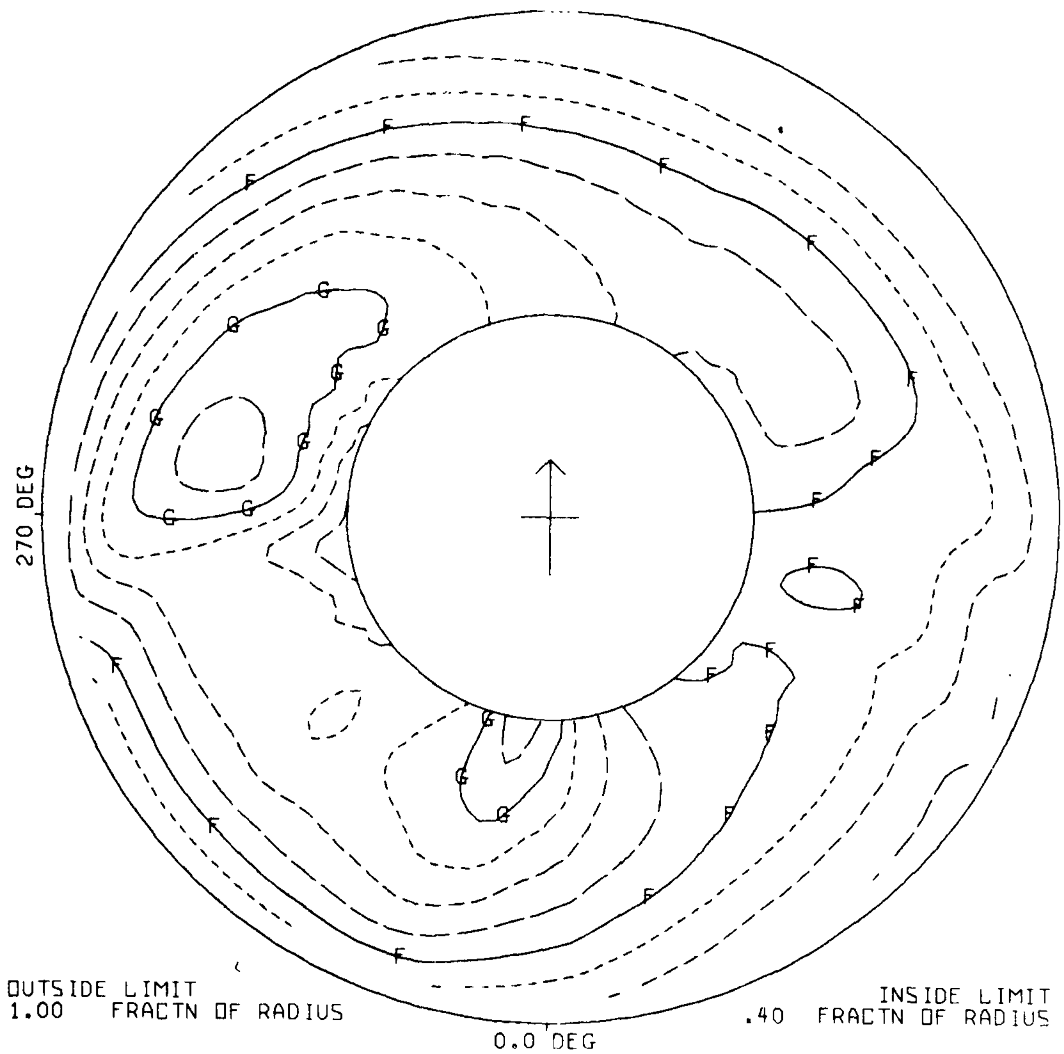
The user now desires to execute this command sequence several times using the Processing Program in the batch mode of execution. After exiting the interactive run of the program using the "TERM/" command, the user must create a data set of card images that supply the proper input for the Processing Program running in batch mode. Following is an appropriate sequence of card images.

```
1
YES
C81COMPR
DERI/TAS,,,1.0208 3.33334/611 0 1/PRINT/
EXECUTE/CNPL %611 %72 %.1 %-1.6/
EXECUTE/CNPL %612 %86 %.1 %-1.6/
EXECUTE/CNPL %613 %101 %.1 %-1.6/
EXECUTE/CNPL %614 %115 %.1 %-1.6/
EXEC/CNPL %615 %129 %.1 %-1.6/
EXEC/CNPL %610 %143 %.1 %-1.6/
TERM/
```

The first three lines must be related to the Processing Program Initialization Phase as explained in Section 5.1. The "1" entry selects the batch mode of operation, the "YES" entry accepts the initial run settings without change, and the "C81COMPR" entry selects the partition for the first partition access slot. The first command step derives true airspeed and establishes the default airspeed calibration constants. Then the next six commands execute the CNPL command sequence for the counters 611, 612, 613, 614, 615, and 610. The counter sequence is rearranged so that the airspeeds will appear in ascending order. Finally, a "TERM/" command ends program execution. Notice that the same scale output values are specified for the contour plot for each execution of "CNPL." This is done so that the resultant plots will have the same scale and thus will be easy to compare. The plot output is generated on the designated device for DATAMAP batch graphic output. These plots are shown in Figures 52 through 57.

5.7.5 Computing the Frequency Response Function

No pair of functions in the OLS data base satisfy the requirement of being the input and corresponding output for a linear system. However, the digital filtering algorithm available in DATAMAP is a linear system so that input to and output from this analysis can be used to illustrate the Frequency Response Function analysis.



LEVEL FLIGHT AT 72 KNOTS OLS DATA

DERIVED PARAMETER: NORMAL FORCE COEFFICIENT

COUNTER 611 GROSS WT 8300 SHIP MODEL AH-1G

LONG CG 200.6 SHIP ID 20391

--------------- CONTOUR LEVEL VALUES IN CN ---------------

A	-1.6	E	.0
B	-1.2	F	.4
C	-.8	G	.8
D	-.4	H	1.2

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/19/80

Figure 52. C_n contour plot for 72 knots.

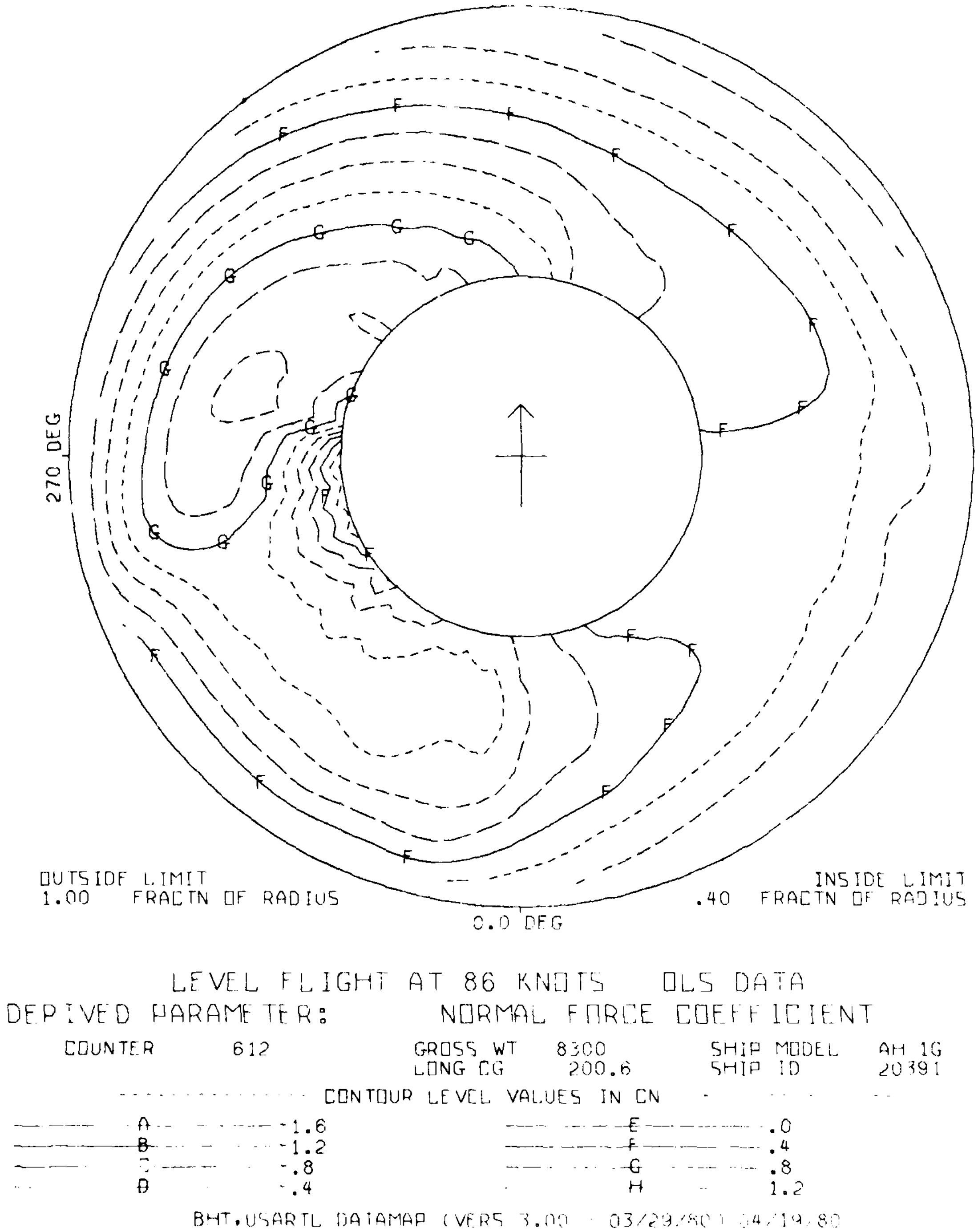


Figure 53. C_n contour plot for 86 knots.

AD-A095 188 BELL HELICOPTER TEXTRON FORT WORTH TX F/G 9/2
THE DATA FROM AEROMECHANICS TEST AND ANALYTICS -- MANAGEMENT AN--ETC(U)
DEC 80 R B PHILBRICK DAAK51-79-C-0015
UNCLASSIFIED BHT-699-099-025-VOL-1 USAAVRADCOM-TR-80-D-30A NL

3 OF 3
AD
A095188

END
DATE
FILMED
3-81
DTIC

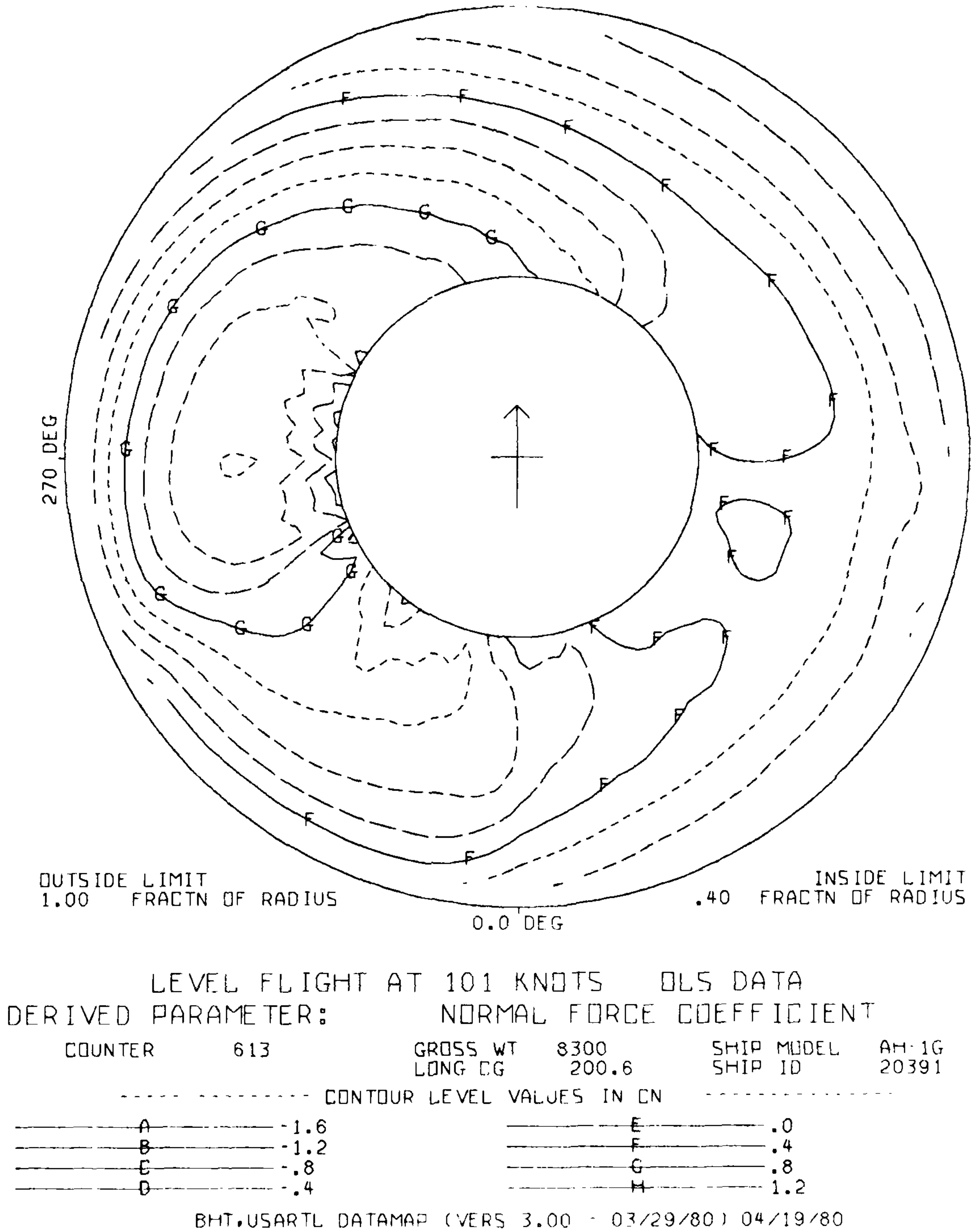


Figure 54. C_n contour plot for 101 knots.

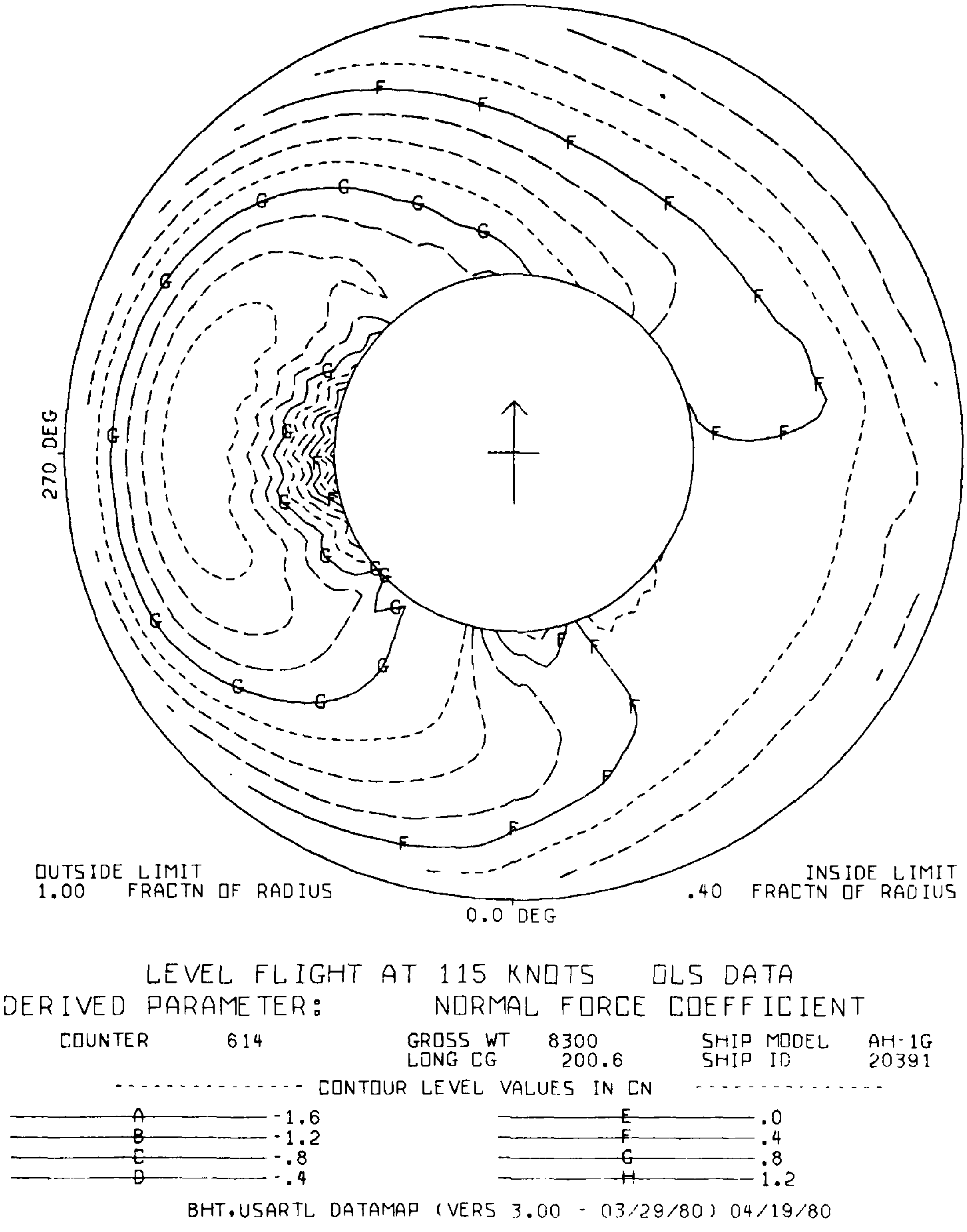
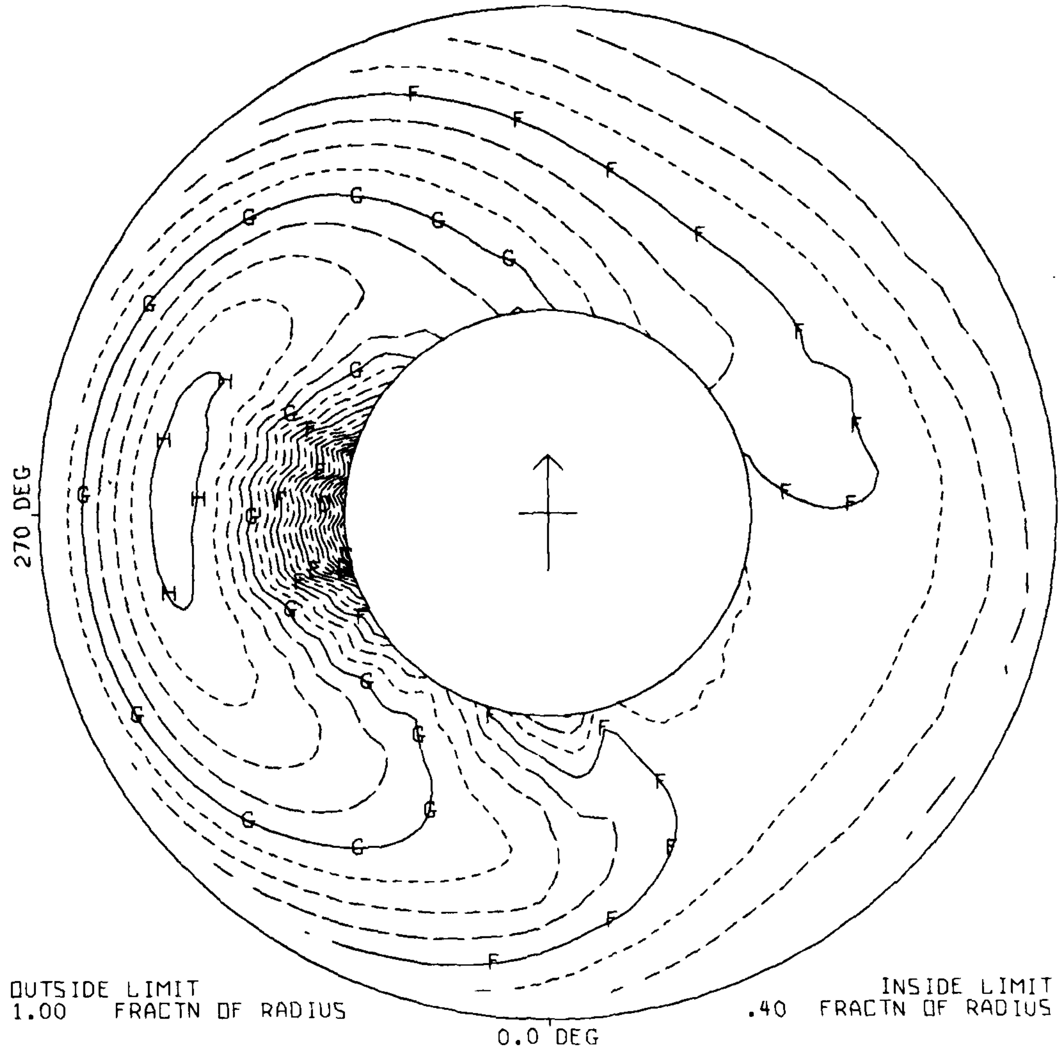


Figure 55. C_n contour plot for 115 knots.



LEVEL FLIGHT AT 129 KNOTS OLS DATA

DERIVED PARAMETER: NORMAL FORCE COEFFICIENT

COUNTER	615	GROSS WT	8300	SHIP MODEL	AH-1G
		LONG CG	200.6	SHIP ID	20391

CONTOUR LEVEL VALUES IN CN

A	-1.6	E	.0
B	-1.2	F	.4
C	-.8	G	.8
D	-.4	H	1.2

BHT.USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/19/80

Figure 56. C_n contour plot for 129 knots.

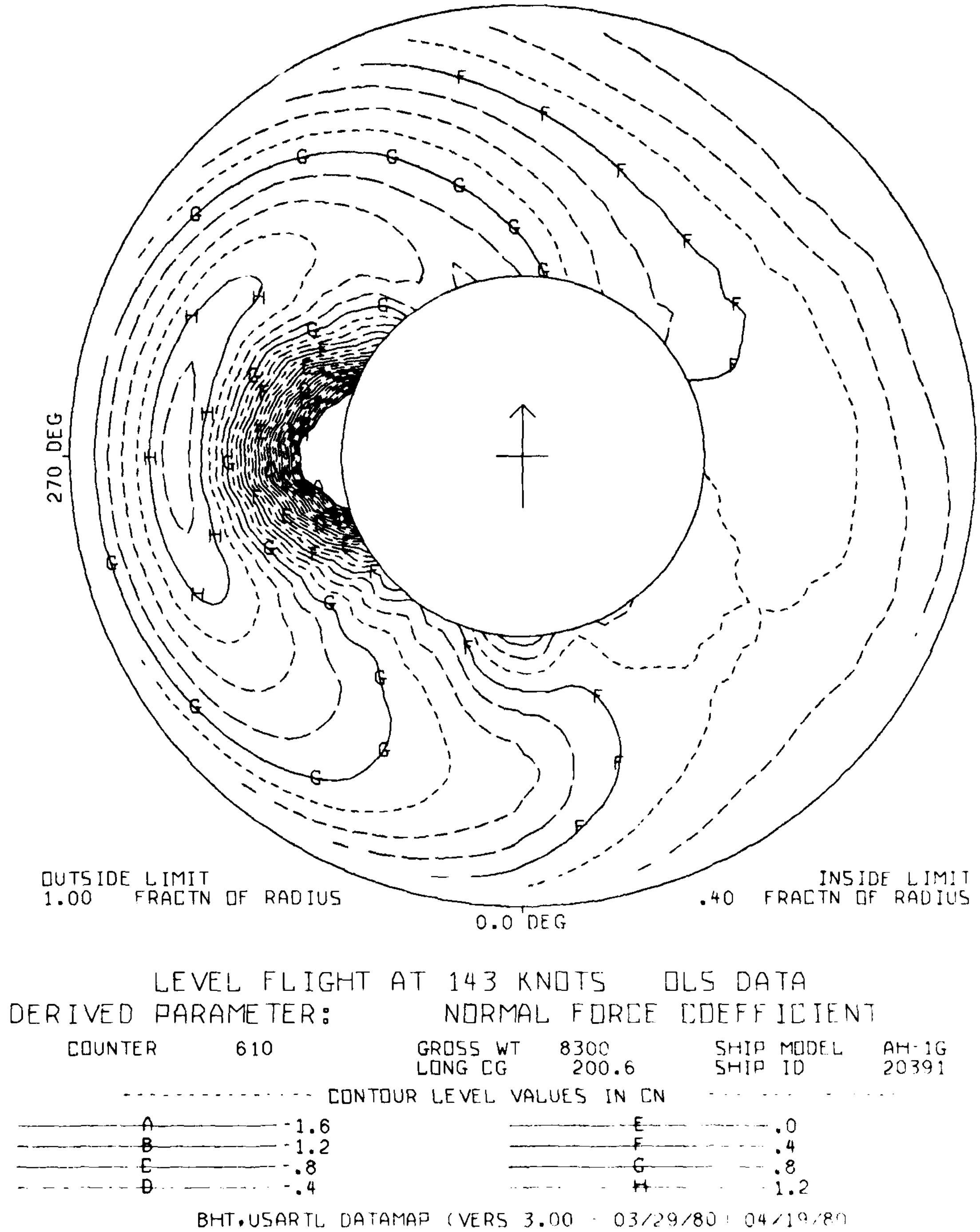


Figure 57. C_n contour plot for 143 knots.

The Processing Program is entered in interactive graphics mode and a partition is accessed that contains acoustic data. These data have a sample rate of 32768 samples per second. First, an Auto-Spectral Density analysis is performed upon some of the data so that the input spectrum will be familiar.

```
 NEW STEP
ANAL/AUTO DENSITY,,HALF 3/IND R849 2 0 .25/
PLOT FREQ,,LOG 10 LOG 4/
 EXECUTING
```

Figure 20 is the plot generated by this command. In the Action Substep, the default maximum frequency is taken (i.e., the Nyquist Frequency for the data), the Half-Cosine Window is selected, and an adjacent point averaging factor of "2" is specified. An adjacent point averaging factor of 2 means that each value in the output will be averaged from five adjacent points and assigned to the frequency position of the central point. In the Input Substep, INDIVIDUAL is selected for the ensemble averaging mode. This indicates that a single time history is input and no ensemble averaging is performed. The balance of the Input Substep is standard. "2" is the counter for the acoustic data. One quarter second of data is selected for processing because no more data will fit in the program storage area at the sample rate of 32768. In the Disposition Substep, logarithmic scales are selected for both the "X" and the "Y" axes with ten decades specified for the "Y" axis and four decades specified for the "X" axis.

To set up input for the Frequency Response Function, the input and corresponding output time histories must be loaded on scratch files. In this case, the input data are loaded to a scratch file and then the corresponding output stored on another scratch file during the filtering step. For the Frequency Response Function, the input record length is limited to 1/8 second because two records are processed simultaneously.

```
 NEW STEP DISP/R849 2 0 .125/KEEP SCF1/
 EXECUTING

 NEW STEP
ANAL/FILT 1000 500 7/SCF1/KEEP SCF2/
 EXECUTING
```

The result of these two commands is that an input time history is loaded in scratch file one and a corresponding output time history is loaded in scratch file two. Now Frequency Response Function analysis can be performed on these time histories and the results displayed. First, the user enters an appropriate comment to describe the filter.

```
 NEW STEP
COMM/SEVEN POLE BAND PASS FILTER 500 to 1000 HZ/

 NEW STEP
ANAL/RESPONSE 2000 HALF 1/IND SCF1 SCF2/PLOT/
 EXECUTING
```

Figure 58 is the result of these two commands. The selection of individual processing, "IND", over ensemble averaging is not significant in this example bacause only one pair of time histories is processed. The accuracy of Frequency Response analysis can be substantially improved by the use of ensemble averaging. An appropriate "ensemble" can be created by loading the scratch files with data that correspond to successive, contiguous time intervals.

```
 NEW STEP
DISP/R849 2 .125 .125/ADD SCF1/
 EXECUTING

 NEW STEP
DISP/R849 2 .250 .125/ADD SCF1/
 EXECUTING

 NEW STEP
DISP/R849 2 .375 .125/ADD SCF1/
 EXECUTING

 NEW STEP
DISP/R849 2 .500 .125/ADD SCF1/
 EXECUTING

 NEW STEP
DISP/R849 2 .625 .125/ADD SCF1/
 EXECUTING

 NEW STEP
DISP/R849 2 .750 .125/ADD SCF1/
 EXECUTING

 NEW STEP
DISP/R849 2 .875 .125/ADD SCF1/
 EXECUTING
```

Now eight time histories are loaded in scratch file one corresponding to one second of data broken into eight contiguous segments. The corresponding filter output time histories are easily obtained.

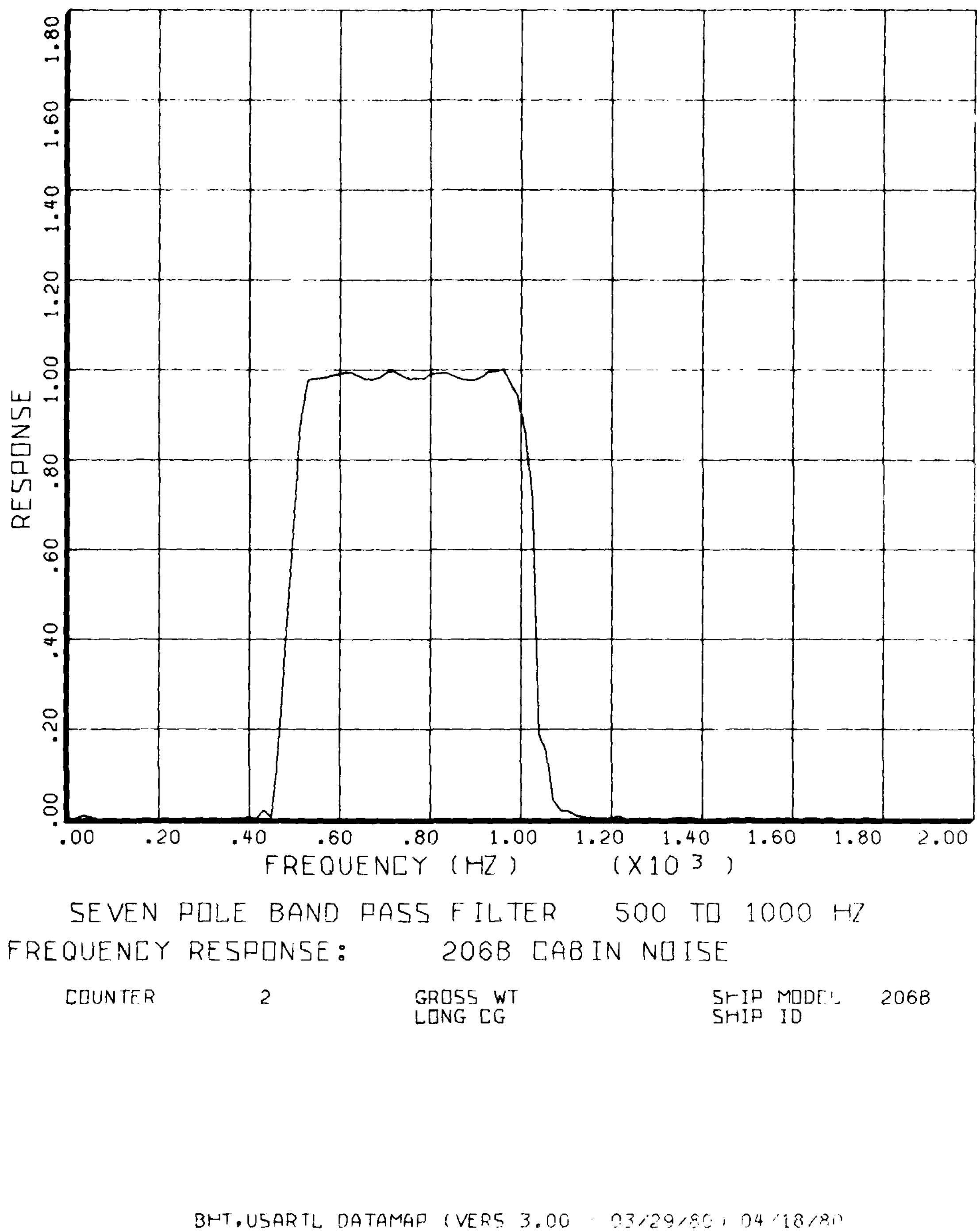


Figure 58. Frequency response function for band pass filter.

```
 NEW STEP
ANAL/FILT 500 1000 7/SCF1/KEEP SCF2/
 EXECUTING
```

Now the ensemble averaged Frequency Response Function can be obtained.

```
 NEW STEP
ANAL/RESP 2000 HALF 1/ENSEMBLE SCF1 SCF2/PLOT/
 EXECUTING
```

Figure 59 is the plot that results from this command. The ensemble averaged function is somewhat smoother and more regular than the individual time history function in Figure 58. To obtain the phase part of the Frequency Response Function, the "DPLOT" output option should be used.

```
 NEW STEP
ANAL/RESP 2000 HALF 1/ENS SCF1 SCF2/DPLOT/
 EXECUTING
```

Figure 21 shows the plot drawn by this command. Notice that the phase of the input is undistorted in the passband of the filter, while the phase of the input is considerably distorted outside the passband.

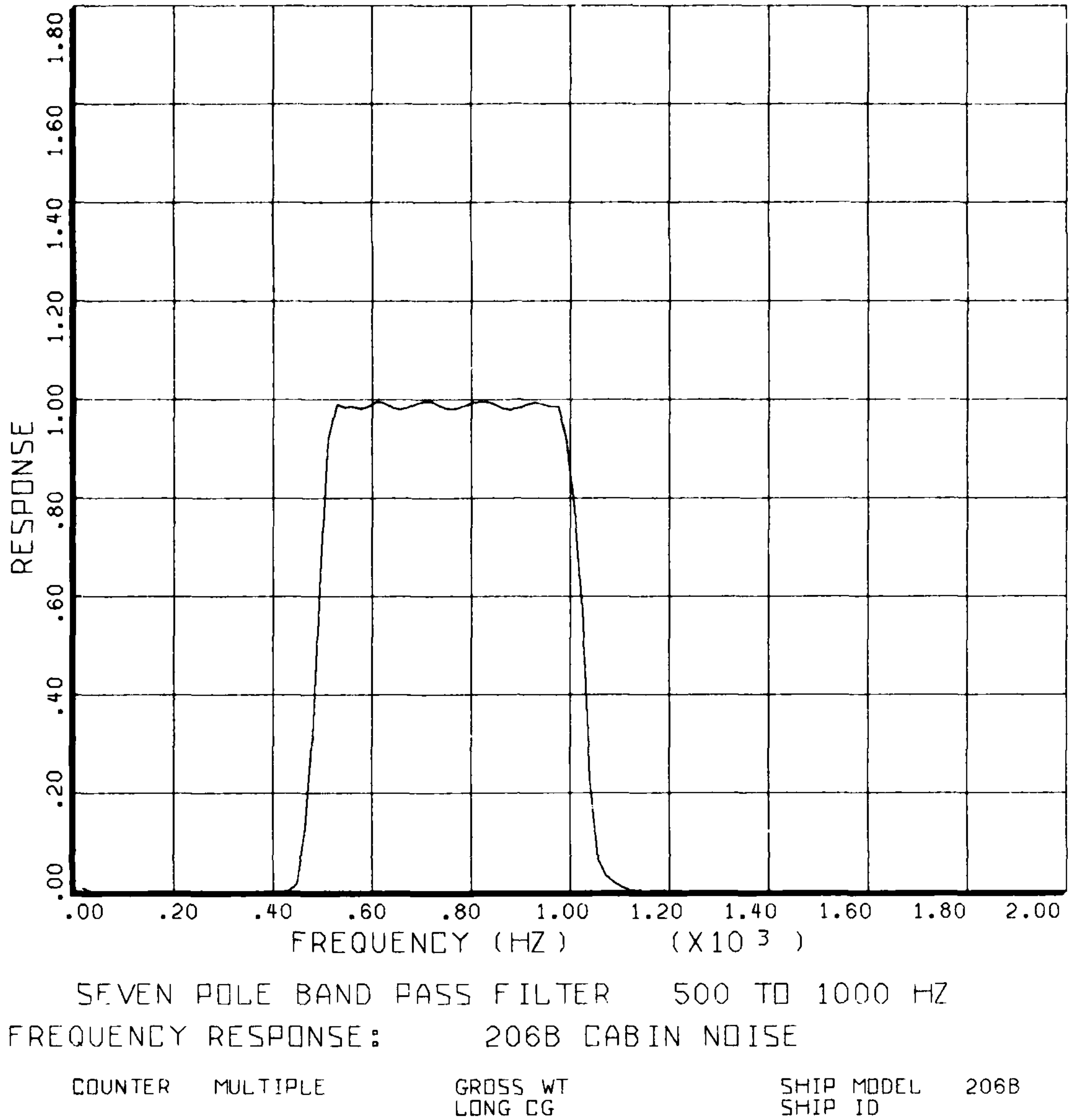
The data in scratch files one and two may be used to illustrate the Coherence Function. Ensemble averaging must be used for this analysis.

```
 NEW STEP
ANAL/COHERENCE 2000 HALF 1/SCF1 SCF2/PLOT/
 EXECUTING
```

The above command generates the plot in Figure 60. The level of coherence or correlation is perfect or nearly perfect in the passband and is very poor outside the passband. Notice that in the Input Substep of the above command, no ENSEMBLE entry is required. Ensemble averaging is required for Coherence Function analysis so there is no entry option to select ENSEMBLE or INDIVIDUAL.

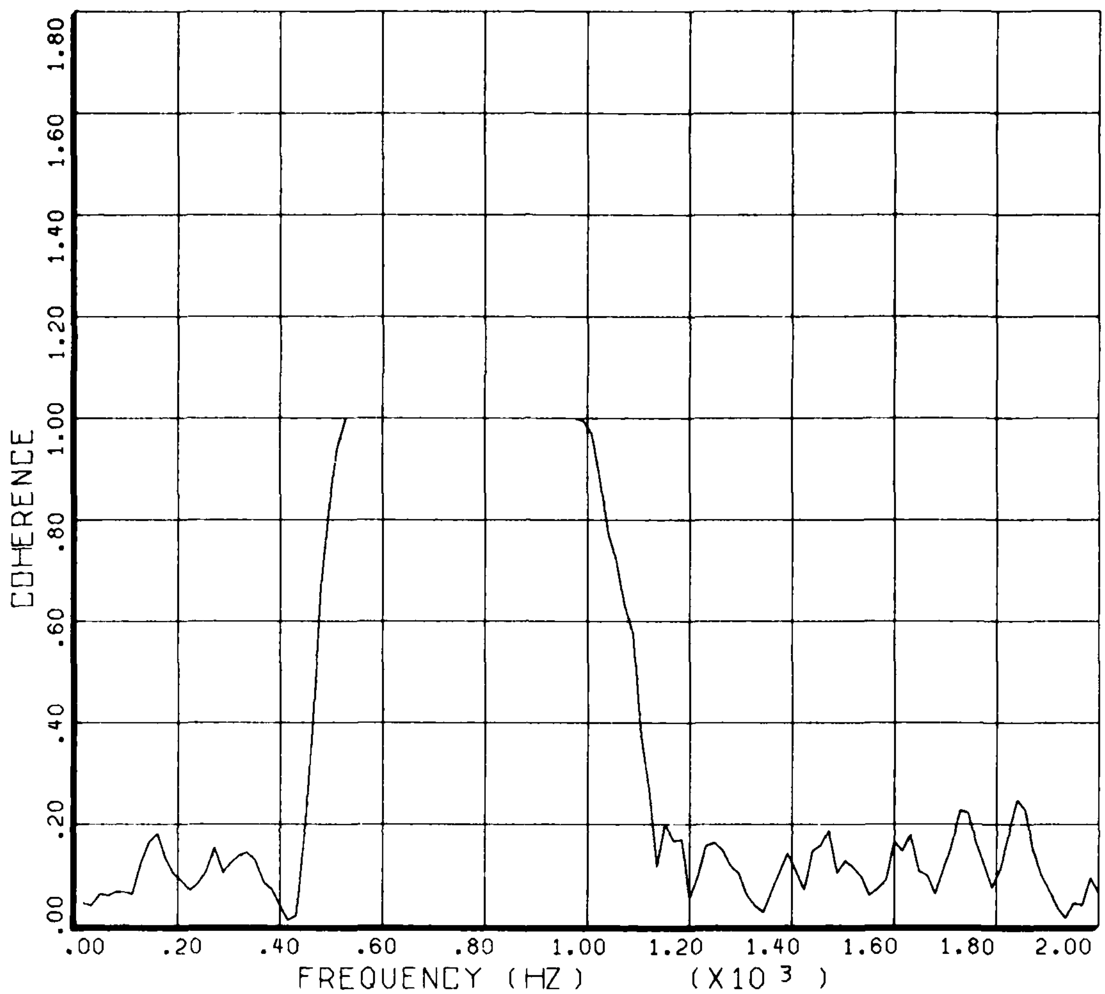
5.8 INFO FILE FORMAT

The Info File is provided as a method to simplify user specification of large numbers of item codes representing like sensors, geometric positions of sensors, and labeling information for output of processes using data from these sensors. The file can be maintained in the same way that program source decks and formatted data decks are maintained. For the OLS application, a specific Info File is provided. However, the



BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/18/80

Figure 59. **Ensemble-averaged frequency response function for band-pass filter.**



SEVEN POLE BAND PASS FILTER 500 TO 1000 HZ
COHERENCE FUNCTION: 206B CABIN NOISE

COUNTER MULTIPLE GROSS WT SHIP MODEL 206B
LONG CG SHIP ID

BHT,USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/18/80

Figure 60. Coherence function for band-pass filter input/output.

user is free to create a new Info File or modify the existing OLS Info File for a special application. Naturally, the user must stay within the specified format for the Info File and, for certain derivations, must follow processing conventions and provide all required information for the processes to be performed.

The Info File consists of the <u>initial group</u> and one or more <u>geometric groups</u>. A geometric group specifies like item codes, arranges those item codes in rows, columns and double-row elements as necessary, specifies geometric positions for the row and/or column elements, and provides overall labels for the quantity the item codes measure and for the variables represented by the row and column positions. A geometric group is a sequence of contiguous card images in the Info File that begins with a four-character name and ends with a line containing the single entry "END."

The initial group consists of a sequence of contiguous lines in the Info File that ends with a line containing the single entry "END." The initial group must appear first in the Info File, and no four-character name is provided for this group. The purpose of the initial group is to provide keyword pointers to individual item codes for important, frequently used quantities such as main rotor azimuth. In addition, the initial group may specify conversion of dependent-variable units on output from the program. Figure 61 lists the OLS Info File
14 and can be used as a general example. No unit conversions are
F specified in this particular Info File. Overall, an Info File consists of the initial group followed by the geometric groups. No lines should be included in a Info File that are not part of a group.

5.8.1 <u>Format for Initial Group</u>

The first section of the initial group consists of keyword pointers and corresponding item codes. Each keyword must be four characters in length and must be the first entry on a new line. A keyword is followed by as many as five item codes. Thus, a keyword signifies a particular quantity and the item codes indicate channels that measure that quantity. It is desirable that only one of the listed item codes following a keyword be present on a partition for each counter. Each item code may be immediately followed by a numeric entry. The meaning of this entry depends upon the keyword and the number is supplied to a derivation algorithm. For rotor azimuth, the number is the true azimuth for a nominal data indication of zero degrees azimuth. No number should be provided if it is not required. Currently, numbers should only follow main and tail rotor azimuth item codes. For example, in Figure 61 the static pressure item code is not followed by a

```
MRAZ R992 338., R1Ø6 338., RØ18 3Ø7.68/
TRAZ RØ25 45.Ø/
TIAS PØØ2/
OATM TØØ4/
STAT PØ3Ø/
MTOR M1Ø7/
END
S1BV  BLADE BEAMWISE VIBRATION
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.2273,.3Ø87,.39Ø2,.5ØØØ,.59Ø2,.7ØØØ,.9Ø2Ø,.9962//
BLBV//
A938/A939/A94Ø/A95Ø/A951/A952/A953/A954//
END
S1CV  BLADE CHORDWISE VIBRATION
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.2273,.3Ø87,.39Ø2,.5ØØØ,.59Ø2,.7ØØØ,.9Ø2Ø,.9962//
BLCV//
A955/A956/A967/A968/A969/A97Ø/A971/A972//
END
S2BU  BL BUTTONS   UPPER SURFACE
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.4Ø,.6Ø,.75,.864,.955//
FRACTN OF CHORD
X/CHORD
LEADING EDGE
.3Ø,.6Ø,.9Ø//
BLUI,BLUO//
INBOARD POINTING
OUTBOARD POINTING
P758,.999,P759,-.985/P77Ø,.862,P771,-.918/
P75Ø,-.934,P751,.949/P732,.893,P733,-.959/
P982,-.949,P983,.922//
P76Ø,-.886,P761,.992/P772,.945,P773,-.976/
P752,-.934,P753,1.ØØ7/P734,.937,P735,.925/
P984,.971,P985,-1.ØØ4//
P762,-.881,P763,-1.Ø58/P741,-.94Ø,P742,.869/
P754,.768,P725,-.825/P736,-.9Ø5,P737,.946/
P986,-.992,P987,1.ØØ4//
END
S2BL  BL BUTTONS   LOWER SURFACE
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.4Ø,.6Ø,.75,.864,.955//
FRACTN OF CHORD
X/CHORD
LEADING EDGE
.3Ø,.6Ø,.3Ø//
BLLI,BLLO//
INBOARD POINTING
OUTBOARD POINTING
P764,-.966,P765,.982/P743,.778,P744,-.799/
P726,-.998,P727,-1.1Ø9/P976,-.941,P977,-1.ØØ6/
P988,-.98Ø,P964,1.Ø36//
P766,-.877,P767,.921/P745,.772,P746,-.939/
P728,-.918,P729,.865/P978,.98Ø,P979,-1.Ø25/
P965,-.944,P966,.959//
P768,.89Ø,P769,-.845/P747,.825,P748,-.935/
P73Ø,-.943,P731,.976/P98Ø,.99Ø,P981,-.983/
P755,-.941,P756,.967//
END
```

Figure 61. OLS Info File.

```
S2HW  HOT-WIRE ATTENUATION SENSORS
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.4ØØØ,.5924,.75ØØ,.8639,.9545//
CONTOUR POSITION
INCHES
LEADING EDGE
-1.56,-1.44,-1.32 -1.2Ø,-1.Ø8,-Ø.96,-Ø.84,-Ø.72,
-Ø.6Ø,-Ø.48,-Ø.36,-Ø.24,-Ø.12,Ø.Ø,Ø.12,Ø.24,Ø.36//
HWAT//
NULL/V171/V82Ø/V866/V923//NULL/V172/V821/V867/V924//
NULL/V183/V832/V878/V925//NULL/V184/V833/V879/V929//
NULL/V185/V834/V88Ø/V93Ø//NULL/V186/V835/V881/V931//
NULL/V8ØØ/V846/V894/V932//V151/V8Ø1/V847/V895/V933//
V152/V8Ø2/V848/V896/V934//V153/V8Ø3/V849/V897/V944//
V154/V8Ø4/V85Ø/V898/V945//V155/V8Ø5/V851/V899/V946//
V156/V816/V862/V911/V947//V167/V817/V863/V913/V949//
V168/V818/V864/V915/V961//V169/V819/V865/V922/V963//
V17Ø/NULL/NULL/NULL/NULL//
END
S1BB  BLADE BEAMWISE BENDING
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.Ø227,.2273,.3Ø87,.39Ø2,.5ØØØ,.59Ø2,.7ØØØ,.8Ø42,.9Ø2Ø//
BLBB//
B112/B12Ø/B126/B128/B122/B13Ø/B132/B124/B134//
END
S1CB  BLADE CHORDWISE BENDING
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.Ø227,.Ø436,.2273,.3Ø87,.39Ø2,.5ØØØ,.59Ø2,.7ØØØ,.8Ø42,.9Ø2Ø//
BLCB//
B113/B115/B121/B127/B129/B123/B131/B133/B125/B135//
END
S1BT  BLADE TORSION
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.Ø227,.3Ø87,.5ØØØ,.7ØØØ,.9Ø2Ø//
BLTR//
M9Ø6/M15Ø/M935/M936/M937//
END
S2HS  HOT-WIRE ATTENUATION - SPECIAL
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.5924,.75ØØ,.8639,.9545//
CONTOUR POSITION
INCHES
LEADING EDGE
-1.56,-1.44,-1.32,-1.2Ø,-1.Ø8,-Ø.96,-Ø.84,-Ø.72,
-Ø.6Ø,-Ø.48,-Ø.36,-Ø.24,-Ø.12,Ø.Ø,Ø.12,Ø.24//
HWAT//
V171/V82Ø/V866/V923//V172/V821/V867/V924//
V183/V832/V878/V925//V184/V833/V879/V929//
V185/V834/V88Ø/V93Ø//V186/V835/V881/V931//
V8ØØ/V846/V894/V932//V8Ø1/V847/V895/V933//
V8Ø2/V848/V896/V934//V8Ø3/V849/V897/V944//
V8Ø4/V85Ø/V898/V945//V8Ø5/V851/V899/V946//
V816/V862/V911/V947//V817/V863/V913/V949//
V818/V864/V915/V961//V819/V865/V922/V963//
END
```

Figure 61. OLS Info File (Continued)

```
S2PP  BLADE ABSOLUTE PRESSURE
AZIMUTH 18Ø.Ø
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.4Ø,.6Ø,.75,.864,.955//
FRACTN OF CHORD
X/CHORD
LEADING EDGE
.ØØ9991,.Ø29972,.Ø7993Ø,.149869,.199825,.249782,.349694,
.399651,.4496Ø7,.499563,.54952Ø,.599476,.699389,.919196//
BLAP,BLAM//
TOP SURFACE
BOTTOM SURFACE
P157,.Ø16697,P173,-.Ø16697/P187,.Ø16697,P8Ø9,-.Ø16697/
P828,.Ø16697,P856,-.Ø16697/P164,.Ø16697,P831,-.Ø16697/
P9Ø8,.Ø16697,P958,-.Ø16697//
P158,.Ø26953,P174,-.Ø26953/P188,.Ø26953,P81Ø,-.Ø26953/
P836,.Ø26953,P857,-.Ø26953/P165,.Ø26953,P843,-.Ø26953/
P9Ø9,.Ø26953,P959,-.Ø26953//
P159,.Ø3912Ø,P175,-.Ø3912Ø/P189,.Ø3912Ø,P811,-.Ø3912Ø/
P837,.Ø3912Ø,P858,-.Ø3912Ø/P166,.Ø3912Ø,P844,-.Ø3912Ø/
P919,.Ø3912Ø,P973,-.Ø3912Ø//
NULL,.Ø46362,NULL,-.Ø46362/P19Ø,.Ø46362,P812,-.Ø46362/
P838,.Ø46362,P868,-.Ø46362/P18Ø,.Ø46362,P845,-.Ø46362/
P92Ø,.Ø46362,P974,-.Ø46362//
NULL,.Ø48165,NULL,-.Ø48165/NULL,.Ø48165,NULL,-.Ø48165/
P839,.Ø48165,P869,-.Ø48165/P181,.Ø48165,P859,-.Ø48165/
NULL,.Ø48165,NULL,-.Ø48165//
P16Ø,.Ø48164,P176,-.Ø48164/P191,.Ø48164,P822,-.Ø48164/
P84Ø,.Ø48164,P87Ø,-.Ø48164/P182,.Ø48164,P86Ø,-.Ø48164/
P921,.Ø48164,P975,-.Ø48164//
NULL,.Ø44446,NULL,-.Ø44446/P192,.Ø44446,P823,-.Ø44446/
P841,.Ø44446,P871,-.Ø44446/P194,.Ø44446,P861,-.Ø44446/
P926,.Ø44446,P989,-.Ø44446//
NULL,.Ø41355,NULL,-.Ø41355/NULL,.Ø41355,NULL,-.Ø41355/
P842,.Ø41355,P872,-.Ø41355/P195,.Ø41355,P875,-.Ø41355/
P927,.Ø41355,P99Ø,-.Ø41355//
P161,.Ø38Ø71,P177,-.Ø38Ø71/P193,.Ø38Ø71,P824,-.Ø38Ø71/
P852,.Ø38Ø71,P873,-.Ø38Ø71/P196,.Ø38Ø71,P876,-.Ø38Ø71/
P928,.Ø38Ø71,P991,-.Ø38Ø71//
NULL,.Ø34788,NULL,-.Ø34788/NULL,.Ø34788,NULL,-.Ø34788/
NULL,.Ø34788,NULL,-.Ø34788/P813,.Ø34788,P877,-.Ø34788/
P941,.Ø34788,P738,-.Ø34788//
NULL,.Ø315Ø4,NULL,-.Ø315Ø4/P8Ø6,.Ø315Ø4,P825,-.Ø315Ø4/
P853,.Ø315Ø4,P874,-.Ø315Ø4/P814,.Ø315Ø4,P891,-.Ø315Ø4/
P942,.Ø315Ø4,P739,-.Ø315Ø4//
NULL,.Ø2822Ø,NULL,-.Ø2822Ø/NULL,.Ø2822Ø,NULL,-.Ø2822Ø/
NULL,.Ø2822Ø,NULL,-.Ø2822Ø/P815,.Ø2822Ø,P892,-.Ø2822Ø/
NULL,.Ø2822Ø,NULL,-.Ø2822Ø//
P162,.Ø21653,P178,-.Ø21653/P8Ø7,.Ø21653,P826,-.Ø21653/
P854,.Ø21653,P884,-.Ø21653/P829,.Ø21653,P893,-.Ø21653/
P943,.Ø21653,P74Ø,-.Ø21653//
P163,.ØØ72Ø5,P179,-.ØØ72Ø5/P8Ø8,.ØØ72Ø5,P827,-.ØØ72Ø5/
P855,.ØØ72Ø5,P885,-.ØØ72Ø5/P83Ø,.ØØ72Ø5,P9Ø7,-.ØØ72Ø5/
P957,.ØØ72Ø5,P757,-.ØØ72Ø5//
END
```

Figure 61. OLS Info File (Continued).

```
S2PX  BLADE ABSOLUTE PRESSURE
AZIMUTH 18Ø.
FRACTN OF RADIUS
R/RADIUS
BLADE ROOT
.4Ø,.75,.864,.955//
FRACTN OF CHORD
X/CHORD
LEADING EDGE
.ØØ9991,.Ø29972,.Ø7993Ø,.149869,.199825,.249782,.349694,
.399651,.4496Ø7,.499563,.54952Ø,.599476,.699389,.919196//
BLAP,BLAM//
UPPER SURFACE
BOTTOM SURFACE
P157,.Ø16697,P173,-.Ø16697/
P828,.Ø16697,P856,-.Ø16697/P164,.Ø16697,P831,-.Ø16697/
P9Ø8,.Ø16697,P958,-.Ø16697//
P158,.Ø26953,P174,-.Ø26953/
P836,.Ø26953,P857,-.Ø26953/P165,.Ø26953,P843,-.Ø26953/
P9Ø9,.Ø26953,P959,-.Ø26953//
P159,.Ø3912Ø,P175,-.Ø3912Ø/
P837,.Ø3912Ø,P858,-.Ø3912Ø/P166,.Ø3912Ø,P844,-.Ø3912Ø/
P919,.Ø3912Ø,P973,-.Ø3912Ø//
NULL,.Ø46362,NULL,-.Ø46362/
P838,.Ø46362,P868,-.Ø46362/P18Ø,.Ø46362,P845,-.Ø46362/
P92Ø,.Ø46362,P974,-.Ø46362//
NULL,.Ø48165,NULL,-.Ø48165/
P839,.Ø48165,P869,-.Ø48165/P181,.Ø48165,P859,-.Ø48165/
NULL,.Ø48165,NULL,-.Ø48165//
P16Ø,.Ø48164,P176,-.Ø48164/
P84Ø,.Ø48164,P87Ø,-.Ø48164/P182,.Ø48164,P86Ø,-.Ø48164/
P921,.Ø48164,P975,-.Ø48164//
NULL,.Ø44446,NULL,-.Ø44446/
P841,.Ø44446,P871,-.Ø44446/P194,.Ø44446,P861,-.Ø44446/
P926,.Ø44446,P989,-.Ø44446//
NULL,.Ø41355,NULL,-.Ø41355/
P842,.Ø41355,P872,-.Ø41355/P195,.Ø41355,P875,-.Ø41355/
P927,.Ø41355,P99Ø,-.Ø41355//
P161,.Ø38Ø71,P177,-.Ø38Ø71/
P852,.Ø38Ø71,P873,-.Ø38Ø71/P196,.Ø38Ø71,P876,-.Ø38Ø71/
P928,.Ø38Ø71,P991,-.Ø38Ø71//
NULL,.Ø34788,NULL,-.Ø34788/
NULL,.Ø34788,NULL,-.Ø34788/P813,.Ø34788,P877,-.Ø34788/
P941,.Ø34788,P738,-.Ø34788//
NULL,.Ø315Ø4,NULL,-.Ø315Ø4/
P853,.Ø315Ø4,P874,-.Ø315Ø4/P814,.Ø315Ø4,P891,-.Ø315Ø4/
P942,.Ø315Ø4,P739,-.Ø315Ø4//
NULL,.Ø2822Ø,NULL,-.Ø2822Ø/
NULL,.Ø2822Ø,NULL,-.Ø2822Ø/P815,.Ø2822Ø,P892,-.Ø2822Ø/
NULL,.Ø2822Ø,NULL,-.Ø2822Ø//
P162,.Ø21653,P178,-.Ø21653/
P854,.Ø21653,P884,-.Ø21653/P829,.Ø21653,P893,-.Ø21653/
P943,.Ø21653,P74Ø,-.Ø21653//
NULL,.ØØ72Ø5,P179,-.ØØ72Ø5/
P855,.ØØ72Ø5,P885,-.ØØ72Ø5/P83Ø,.ØØ72Ø5,P9Ø7,-.ØØ72Ø5/
P957,.ØØ72Ø5,P757,-.ØØ72Ø5//
END
```

Figure 61. OLS Info File (Concluded).

number since a number is not required for the static pressure derivation. Following is a list of the keywords that are currently recognized by the Processing Program.

MRAZ - Main rotor azimuth using positive pulse encoding

TRAZ - Tail rotor azimuth using positive pulse encoding

MDEG - Main rotor azimuth in degrees

TDEG - Tail rotor azimuth in degrees

TIAS - True indicated airspeed (knots squared)

TASK - True airspeed (knots)

OATM - Outside air temperature (Deg C)

STAT - Static pressure (PSIA)

MTOR - Main rotor mast torque (in.-lb)

TTOR - Tail rotor mast torque (in.-lb)

Each sequence of item codes and numbers that follows a keyword must be terminated by a slash.

After the last keyword line, unit conversion specifications may be entered. If unit conversion for the dependent variable is to be specified, then the next line after the last keyword entry should contain the single entry "UNITS." The next line should contain the single entry "OUTPUT." Then one or more lines specifying unit conversions may follow. Each line should specify one unit conversion. The first entry in a unit conversion line must be a string that duplicates the unit label for the dependent variable in the program immediately before output of the data (i.e., the unit label as currently displayed before the conversion specification is inserted in the Info File). The second entry is the replacement unit label. Either or both of these entries may be enclosed in single quotes (i.e., apostrophes) to avoid line errors and/or unintentional separation of a label with a space or a slash (e.g., 'M/SEC').

Unit conversion of dependent variable values is accomplished using the linear relation

$$Y' = MY + B$$

where Y is a value in the original unit system and Y' is the equivalent value in the new unit system. Frequently, the

dependent axis intercept value, "B", is zero, but the Info File will accommodate non-zero values for "B". The "M" value or slope for the conversion follows the replacement unit label and is in turn followed by the "B" value. If the "B" value is zero, it may be left blank. Figure 62 shows an initial group with several unit conversion specifications.

Whether or not unit conversions are specified, the last line in the initial group must be an "END" entry.

5.8.2 Format for Geometric Groups

The first entry on the first line of a geometric group must be the four-character name for the group. Groups that specify more than one row position are considered to be two-dimensional groups and the first two characters in the name for such a group must be "S2" (e.g., S2PP). Groups that specify a single row position are considered to be one-dimensional and the name for such a group is only required to be a four-character string (e.g., BBM1). Following the group name on the same line is a title for the data specified by the item codes in the group. This label may contain as many as 52 characters beginning with the first non-blank character following the group name. Separators are not considered in the label (i.e., the label is directly copied) but no line errors may be present.

Following the group name and the title line, the next line may specify an azimuth correction angle. If so, the first entry on this line must be the keyword "AZIMUTH" and the second entry must be the correction angle in degrees. This correction angle is added to the overall azimuth correction angle (Modulo 360).

The next line, or the line following the group name and title line if no azimuth correction angle is included, is the label for the column element position variable. This label may contain as many as 16 characters and separators may be included. The next line is a shortened version of the same label that may be as many as eight characters in length. Following is a line that contains a label of as many as 16 characters for some geographic feature that is closest to the smallest column element position. For example, "BLADE ROOT" is frequently used for OLS groups where column element position corresponds to span station. Each of the labels described in this paragraph should begin in column one of the corresponding line. These labels are used to annotate output plots and listings.

Following the column label lines are one or more lines of numeric entries that give column element positions. The numbers should be separated with spaces or commas and two slashes must follow the last number on the line.

```
MRAZ R992 0.0, R106 0.0, R018 30.32/
TRAZ R992 45.0/
TIAS P002/
OATM T004/
STAT P030/
MTOR M107/
UNITS
OUTPUT
PSIA 'KN/M**2' 6.8947572
KNOTS 'M/S' .5144444
FEET METERS .3048
'FEET MSL' METERS .3048
'  HP' WATTS 745.6999
G 'M/S**2' 9.80665
IN M .0254
LB N 4.448222
PSI 'KN/M**2' 6.8947572
END
```

Figure 62. Sample initial group with unit conversion specifications.

For two-dimensional groups, a sequence of lines describing the row element positions follows the column element positions. This sequence must begin on a new line. The format for this description is the same as the description for the columns with three lines of labels followed by one or more lines of row element positions. Values for row element positions must increase monotonically in this listing.

Following the last line of row element positions for two-dimensional groups, or following the last line of column element positions for one-dimensional groups, is a line of one or two four-character keyword entries that describe the data from the item codes in the group. If one keyword is present, only one double-row element is specified in the group. This element is treated as the TOP element. If two keywords are present, then both double-row elements are present and the first keyword corresponds to the TOP element and the second keyword corresponds to the BOTTOM element. The specific keywords that are used are tested by the program in only four instances. "BLAP" and "BLAM" must be used for blade absolute pressure sensors on the top and bottom surfaces respectively to qualify for input to the C_p derivation. "BLBI" and BLBO" must be used for boundary layer button (BLB) sensors pointing inboard and outboard respectively to qualify for input to the local flow direction and local flow magnitude derivations. The line of keywords must be terminated by two slashes.

If two keywords were present in the keyword line, then the following two lines must be labels for the two corresponding double-row elements. The immediately following line applies to the TOP double-row element and the next line applies to the BOTTOM double-row element. These labels must begin in column one and must be no longer than 20 characters. If only one keyword was present in the keyword line, then no double-row element labels should be present.

Following the line containing the BOTTOM double-row element label, or following the keyword line if only one keyword was present, are the lines that specify item codes for the group. All of the item codes for a row must appear in a contiguous sequence that will be called a <u>sentence</u>. A sentence must begin on a new line and may encompass one or more lines. A sentence is composed of one or more <u>phrases</u> that each specify the item code(s) for a column position. The first entry in a phrase is the item code corresponding to the TOP double-row element. A numeric entry may follow this item code. Currently, this number has meaning only for the boundary layer button sensor groups and for the blade absolute pressure groups (see Section 6.2). If the group specifies only one double-row element, then the phrase is complete. If both double-row elements are specified for the group (as specified by two keywords in

the keyword line), then the item code for the BOTTOM double-row element must be included. This item code must follow the TOP item code or the number following the TOP item code if a number entry was provided. A number may also follow the BOTTOM item code. The phrase is terminated with a single slash. Thus, the sequence of entries in a phrase is:

- Item code for TOP double-row element
- Number (if required)
- Item code for BOTTOM double-row element (only if both double-row elements are specified by the keyword line)
- Number (if required and only if a second item code is included)
- Slash

All of the entries for a phrase must appear contiguously on the same line. More than one phrase may appear on a single line.

The number and sequence of phrases in a sentence must correspond to the number and sequence of column element positions that were listed earlier. A sentence must be terminated by two slashes including the slash that ends the last phrase in the sentence. For two-dimensional groups, the sequence of sentences in the group must correspond to the sequence of row element positions that was listed earlier and the number of sentences must be the same as the number of row element positions listed. For one-dimensional groups, only one sentence should be present.

The above requirements may force the entry of an item code for a row/column/double-row intersection when there are no measured or simulated data for that position. The pseudo item code "NULL" may be entered for such cases.

The "END" entry should appear after the last sentence in a group as the only entry in a new line.

5.8.3 Group Generation by the File Creation Program

As described in Section 3, the information required for creation of one or more Info File geometric groups may be stored in a Data Transfer File (DTF). The File Creation Program will automatically generate a file of one or more Info File geometric groups from this information when the DTF is processed. This collection of geometric groups does not constitute a valid Info File by itself. Instead, these groups must

be appended to an initial group so that all of the necessary elements of an Info File are present. Consult local documentation to determine how the initial group and a collection of automatically generated geometric groups are joined or "concatenated" to each other.

5.8.4 Testing an Info File

The DATAMAP Processing Program has two facilities for testing the structure and format for an Info File. First, the initial group of a file is automatically tested during the Initialization Phase (see Section 5.1). Second, all of the geometric groups in the file are tested the first time that a MENU/INFO/ command is executed during a run of the Processing Program.

5.8.5 Info File Conventions

The standard convention for the OLS blade applcation of DATAMAP has been that column positions in the Info File correspond to span stations and row positions correspond to chord stations. Info File geometric groups generated from C81 through a DTF follow this convention in that column positions represent span stations and multiple row stations are not generated. The blade static pressure, blade normal force coefficient, blade chordwise force coefficient, blade pitching moment coefficient, blade displacement, and blade slope derivations all presume that this convention has been followed. However, the user is free to change the convention for other applications.

An additional convention that is followed for OLS blade applications is that the units for column or span position are normalized to fraction of the full blade radius, and the units for row or chord position are normalized to fraction of the full chord width. The above listed derivations also presume that this convention is followed. Geometric groups generated from C81 through a DTF do not follow this convention. However, C81 generated data are not used as input for any of the above derivations.

6. PROCESSING ALGORITHMS

The purpose of this section is to acquaint the user with the mathematical techniques used by the Processing Program ANALYZE or DERIVE command steps. Methods that are well defined and documented in the mathematical literature are covered with brief descriptions and references to that literature. The specific nature of application of these methods to the data streams addressed by this program will be covered in full in this section.

6.1 ANALYSES

6.1.1 Harmonic Analysis

Numerous texts show that a very broad class of continuous and piecewise continuous functions, x(t), may be represented by infinite series of trigonometric sine and cosine functions over a finite interval, T, as

$$x(t) = \frac{a_0}{2} + \sum_{K=1}^{\infty} \left(a_K \cos \frac{2\pi Kt}{T} + b_K \sin \frac{2\pi Kt}{T}\right) , \qquad (1)$$

where

$$a_K = \frac{2}{T} \int_0^T x(t) \cos \frac{2\pi Kt}{T} \, dt$$

and $\qquad K=0,1,2,\ldots$

$$b_K = \frac{2}{T} \int_0^T x(t) \sin \frac{2\pi Kt}{T} \, dt$$

The series is referred to as a Fourier Series and the coefficients, a_K and b_K, are referred to as Fourier coefficients.

Oftentimes, the series is expressed in a different form as derived below:

$$x(t) = \frac{a_0}{2} + \sum_{K=1}^{\infty} a_K{}^2+b_K{}^2 \left[\frac{a_K}{\sqrt{a_K{}^2+b_K{}^2}} \cos \frac{2\pi Kt}{T} + \frac{b_K}{\sqrt{a_K{}^2+b_K{}^2}} \sin \frac{2\pi Kt}{T} \right]$$

If a_K and b_K are not both zero, there exists a number, ϕ_K, such that

$$\cos \phi_K = \frac{a_K}{\sqrt{a_K^2 + b_K^2}} \text{ and } \sin \phi_K = \frac{b_K}{\sqrt{a_K^2 + b_K^2}}$$

where $-\pi < \phi_K \leq \pi$.

After defining

$$c_K = \sqrt{a_K^2 + b_K^2}$$

we may write

$$x(t) = \frac{a_0}{2} + \sum_{K=1}^{\infty} c_K \left[\cos \phi_K \cos \frac{2\pi Kt}{T} + \sin \phi_K \sin \frac{2\pi Kt}{T} \right]$$

Using the well-known trigonometric identity,

$$\cos (\alpha - \beta) = \cos \alpha \cos \beta + \sin \alpha \sin \beta$$

the series may be written as

$$x(t) = \frac{a_0}{2} + \sum_{K=1}^{\infty} c_K \cos \left(\frac{2\pi Kt}{T} - \phi_K\right) , \qquad (2)$$

where

$$\phi_K = \tan^{-1} \left(\frac{b_K}{a_K} \right)$$

is placed in the proper quadrant by inspection of the signs of b_K and a_K. When $a_K = b_K = 0$, then $\phi_K = 0$ arbitrarily.

Thus, referring to Equation (2), each term in the series is called a harmonic and the harmonic number is K. In addition, the frequency in hertz of each term is given by K/T, while the magnitude or amplitude of each term is given by c_K and the phase is given by ϕ_K.

In practical applications involving signal processing, the data are not known as a continuous function nor is it possible to compute an infinite number of harmonics. Fortunately,

only a few harmonics are of interest in many applications. Text books on numerical analysis show that the function, x(t), may be approximated by finite trigonometric series using a sequence of values of the function at various times, t. In particular, let

$$x_j = x(t_j),$$

where

$$t_j = j\Delta t, \; j=0,1,2...,N-1.$$

Then it can be shown that for M sufficiently large,

$$x_j \cong \frac{a_0}{2} + \sum_{K=1}^{M} \left(a_K \cos\frac{2\pi Kj}{N} + b_K \sin\frac{2\pi Kj}{N}\right), \qquad j=0,1,...,N-1 \tag{3}$$

where

$$a_K = \frac{2}{N} \sum_{j=0}^{N-1} x_j \cos\frac{2\pi Kj}{N}$$

$$K=0,1,2,...,M$$

$$b_K = \frac{2}{N} \sum_{j=0}^{N-1} x_j \sin\frac{2\pi Kj}{N}$$

and $M \le N$

Thus, Equation (3) is analgous to Equation (1). It can also be shown that the trigonometric series given by Equation (3) is the best approximation in the least squares sense to the sequence of values, x_j, over all other trigonometric (sine and cosine) series having no more than M terms. If the number of terms, M, is chosen to be N, the series given by Equation (3) is identically x_j and is called the Discrete Fourier Transform.

The Harmonic Analysis algorithm requires as input a time history with a length and start point corresponding to an integer number of rotor cycles. Fourier components are only calculated

for those Harmonic numbers that correspond to a positive integer multiple of the number of cycles represented in the input data. The output harmonics of this process are scaled as the sequential harmonic components of a single rotor cycle of data. When more than one rotor cycle of input data are provided, this process provides a more accurate estimate of steady-state amplitude. In addition, inaccuracies in rotor azimuth are minimized by processing more than a single rotor cycle.

The computational routine is a recursive algorithm described in Reference 4. This process presumes that frequencies of interest in the data are integral multiples of the rotor speed expressed in revolutions per second.

For output, the sine and cosine harmonic terms are converted to amplitude and phase in degrees. The output independent variable scale is normally displayed as Harmonic number for a single rotor cycle. However, the user has the option to specify frequencv as the independent variable. Frequency may only be specified as the independent variable in the same command step in which the Harmonic analysis is actually calculated.

6.1.2 Digital Filtering

Digital filtering operations in the DATAMAP Processing Program are accomplished using Chebyshev filters. The magnitude characteristic of these filters varies between equal maximum and equal minimum values in the passband and decreases monotonically toward zero outside this frequency band. In particular, the square of the magnitude of the Chebyshev transfer function is defined by

$$H(i\omega)^2 = \frac{1}{1+\delta^2 T_N^2(\omega)},$$

where $T_N(\omega)$ is the Nth degree Chebyshev polynomial, which may be defined by

$$T_N(\omega) = \begin{cases} \text{Cos}(N \cos^{-1}\omega) & , 0 \leq \omega \leq 1 \\ \text{Cosh}(N \cosh^{-1}\omega) & , \omega > 1. \end{cases}$$

[4] A. Ralston and H. Wilf, MATHEMATICAL METHODS FOR DIGITAL COMPUTERS, John Wiley and Sons, New York, 1960, Chapter 24.

The constant δ is the amplitude of the oscillation in the passband, $i = \sqrt{-1}$, ω is in units of angular frequency normalized to unity, and N is the number of poles in the transfer function. In general, the phase characteristics of the Chebyshev filters are undesirable. However, a procedure involving two filtering operations on the data results in an effective filter that is phase free and distortionless.[5] A mathematical analysis of the technique is given in Reference 5. To illustrate the technique and to show the magnitude/phase characteristics of a seven-pole Chebyshev filter, a swept frequency sinusoidal function was generated and filtered. Following the procedure, the filter output was manipulated then filtered again. Using Fast Fourier Transform techniques, together with system input/output relations, the frequency response function (transfer function) was computed and is displayed in Figure 63. The user is provided the flexibility of selecting both low-pass and band-pass filters together with the number of poles to be used.

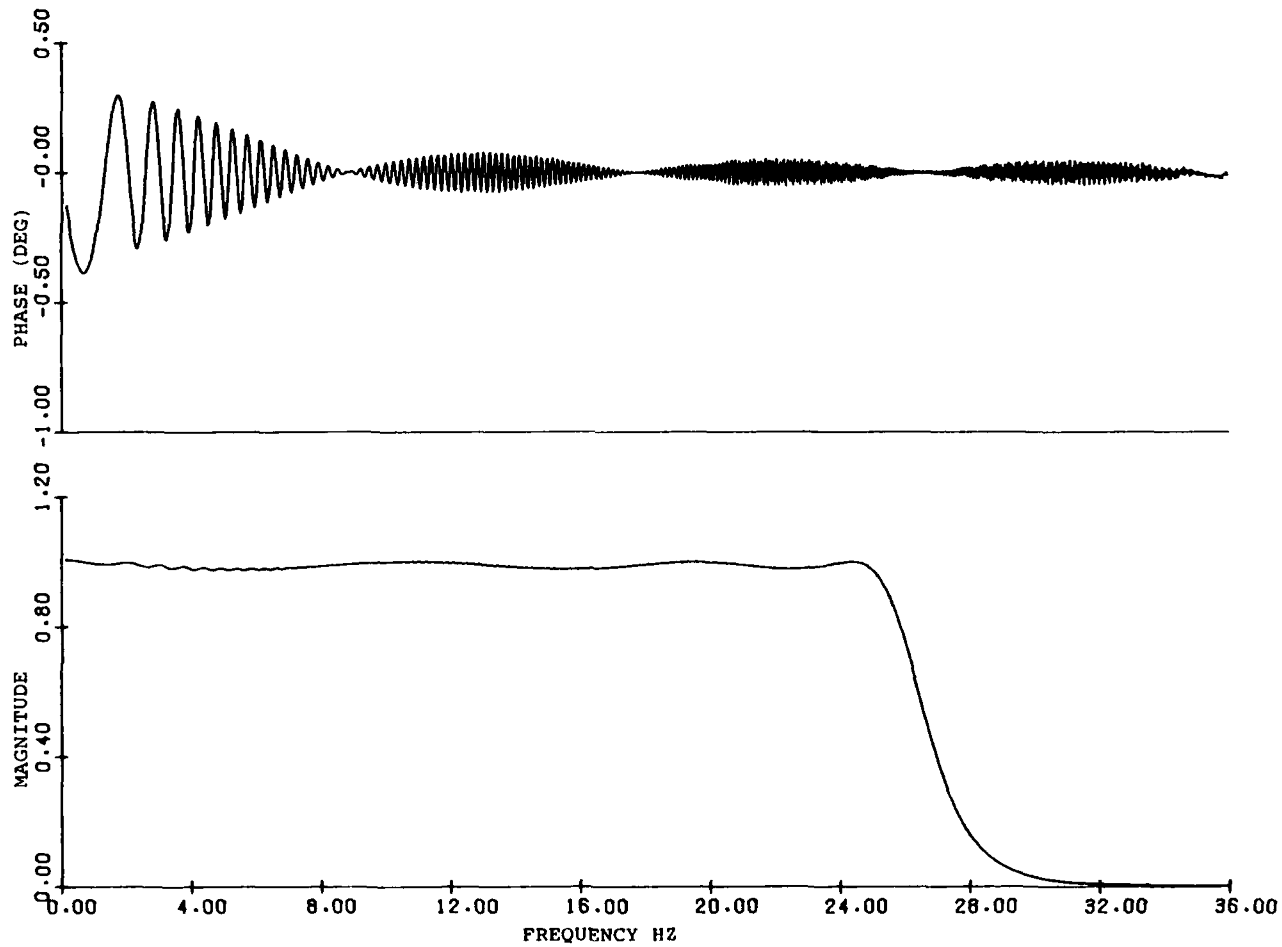
The decrease in magnitude of the filter transfer function with increased frequency above the filter breakpoint is called the <u>rolloff</u> of the filter. Examination of the formulas for the transfer function and the Chebyshev polynomial shows that the rolloff is more rapid for large N and less rapid for small N. The speed with which the transfer function magnitude decreases with increased frequency above the filter breakpoint is called the <u>rate of rolloff</u>.

Figures 48 and 49 show main rotor shaft horsepower for various airspeeds before and after the application of a low-pass digital filter. The filter was specified to have seven poles and a break frequency of 2.5 Hz.

Filtering processes tend to distort finite records near the boundaries of these records. The filtering algorithm seeks to minimize this distortion by synthesizing extensions to the beginning and end of the input data record. Every synthesized data value at the beginning of the data record is set equal to the first real data value. Similarly, every synthesized data value for the end of the data record is set equal to the last real data value. These artificial extensions to the input data record are not retained on output.

---

[5] A. L. Eubanks, FILTER DESIGN AND ANALYSIS WITH APPLICATIONS TO DISCRETE DATA, Bell Helicopter Textron Report 299-099-889, Fort Worth, Texas, 15 August 1977.



25 HERTZ PHASE FREE DIGITAL FILTER

Figure 63. Two pass Chebyshev filter characteristics.

When a complete period of a periodic function is filtered, the edge distortion problem can be particularly acute. The end of the period and the beginning of the next period should be continuous, but the beginning and end of a periodic function frequently do not "match" after digital filtering. Such discontinuities are quite noticeable in cylindrical format contour and surface plots.

In DATAMAP, the filtering process will recognize cycle averaged data, assume the data are periodic, and apply a "circular" filtering algorithm. In this algorithm, the input time history is mapped into a time history twice as long. The first half of the old record is replicated twice as the second and fourth quarters of the new record, and the second half of the old record becomes the first and third quarters of the new record. After the filtering process, the probable distortion occurs at the beginning of the first quarter and the end of the fourth quarter of the filtered new record. The second and third quarters of the filtered new record, which are relatively isolated from boundary effects, become the first and second halves of the filtered output record.

6.1.3 Amplitude Spectra

The Amplitude Spectrum analysis evaluates the amplitude of the Fourier components of an input time history. The algorithm assumes that the time history function is composed of pure Fourier components that are well separated in frequency. The 15
amplitude spectrum technique seeks to provide a means for B
evaluating the magnitude of components at frequencies that are not an integral multiple of the rotor frequency or to evaluate the magnitude of Fourier components when rotor azimuth is not available to specify complete cycles of data input.

The frequency domain representation of an input record is obtained by means of a Fast Fourier Transform (FFT) routine. However, when a finite length data record is Fourier Transformed, the result is a convolution of the 'true' Fourier Transform with the function $C(\omega)$

$$C(\omega) = SIN(L\pi\omega)/L\pi\omega$$

where ω is frequency in Hz, L is the length of the data record in seconds and C is the convolving function. When a frequency component undergoes a complete number of cycles in the data record, the convolution process yields a zero contribution from this component to every evaluated frequency except the proper frequency for the component. However, in the situations that the amplitude spectrum algorithm was designed to handle, most frequency components do not undergo a complete number of cycles in the data record. Frequency components

are "spread" around the correct frequency roughly in proportion to the reciprocal, 1/D, of the separation, D, between a frequency component and the correct frequency. The sum of the squares of these components is equal to the square of the correct frequency component.

The algorithm seeks to reduce this problem by applying a window function to the data record. The effect of a window function is to create a freqency domain convolving function that is shorter and broader than $C(\omega)$ for small D but that decreases much more rapidly than 1/D for large D. The three available windows are cosine taper, hanning and rectangle (see Paragraph 6.1.7). The user may select which of these windows the algorithm will use.

The cosine taper window generally provides the most accurate amplitude evaluations for frequency components that are reasonably well separated. The rectangle window is equivalent to no window at all. This window may provide better results in resolving two frequency components that are very close. The hanning window should be used when adjacent frequency components are well separated but several orders of magnitude different in amplitude. The particular advantage of a window application for the amplitude spectrum algorithm is to create a consistant spread for a pure frequency component whether the component undergoes an integer number of cycles in the sample record or not.

After the window function is applied to the data record, the FFT is performed. The number of points in the input data record is constrained to be a power of two so as to optimize the computational efficiency of the FFT algorithm. After the FFT is performed, the magnitude squared for each frequency component is computed and then a corrective factor is applied to compensate for the effects of the window function. After the corrective factor is applied, the algorithm computes the sums for each set of three adjacent squared magnitude frequency components. Next, the square root of each resultant squared magnitude sum is computed. The result of this process is an amplitude spectrum estimate for which, given the initial assumptions for the algorithm, the peak values displayed are accurate estimates of the amplitude of pure frequency components.

6.1.4 Moving Block Damping Estimation

The Moving Block Damping Analysis algorithm assumes that the input function, or time history, is of the form

$$f(t) = Ae^{-\frac{D\omega t}{100}} \sin(\omega t+\phi) + Q(t),$$

where D is percentage of critical damping, ω is a known or suspected frequency component, Q(t) is a function with frequency components well separated from ω, and ϕ is an arbitrary constant phase. The algorithm extracts a single value, the percentage of critical damping D, for each time history processed.

From the input time history, a sequence of overlapping blocks of data is chosen for analysis such that each block is identical to the preceding block except that the next data sample is included in the block and the first data sample of the preceding block is excluded. Then a Fourier analysis is performed on each block and the logarithm of each magnitude at frequency ω is computed to yield a curve which has a slope equal to the damping associated with the frequency, ω. The principle of least squares is applied to the sequence of log values to obtain the best estimate of damping. Reference 6 reports on the use of the method.

6.1.5 Cycle Averaging

The cycle averaging algorithm seeks to reduce superfluous noise by averaging several contiguous cycles to form a single representative cycle. Initially, 256 equally spaced azimuth positions are established. From each input cycle of data, the computer interpolates values for every established azimuth position. Then the values corresponding to each azimuth position are averaged together.

6.1.6 Min/Max Analysis

Data are evaluated in the sense of min/max through the following process. Input data consisting of an integer number of complete cycles are examined, cycle by cycle, for the minimum and maximum value occurring in each cycle. Then, for every cycle, mean and oscillatory values are computed by

$$\text{Osc} = 1/2\ (\text{max} - \text{min})$$
$$\text{mean} = 1/2\ (\text{max} + \text{min})$$

---

[6] J. G. Yen, S. Viswanathan, and C. G. Matthys, FLIGHT FLUTTER TESTING OF ROTARY WING AIRCRAFT USING A CONTROL SYSTEM OSCILLATION TECHNIQUE, NASA Symposium on Flutter Testing Techniques, Flight Research Center, Edwards AFB, Calif, October 9-10, 1975.

6.1.7 Auto- and Cross-Spectral Density

Formally, the auto-spectral density of a stochastic process is the Fourier Transform of the auto-correlation of the same process, and the cross-spectral density of two stochastic processes is the Fourier Transform of the cross-correlation of the same two processes.[7] DATAMAP calculates an estimate of auto-spectral density from one or more sample input time history records, and estimates cross-spectral density from one or more pairs of sample input time history records. Auto-spectral density is similar to the amplitude spectrum analysis (see Paragrpah 6.1.3) in that a function of time is presented as a function of frequency using Fourier Transform techniques. However, the amplitude spectrum is intended to derive the amplitudes of one or more pure frequency components in a time history. The auto-spectral density is intended to derive the distribution of variance as a function of frequency for a non-periodic process. Cross-spectral density analysis is provided for completeness. Generally, cross-spectral density is an intermediate step in the derivation of more useful functions such as the Coherence Function and the Frequency Response Function (see Paragraphs 6.1.8 and 6.1.9).

Following are the eight steps required for the Auto-Spectral Density calculation.

(1) Adjust the record length as necessary to a power of two samples.

(2) Reset the mean level for the input data to zero.

(3) Apply the window function selected by the user.

(4) Perform the Fast Fourier Transform computation.

(5) Calculate the magnitude squared for each complex frequency component.

(6) Compute and apply a constant factor to adjust for the record length, sample interval, and window function.

---

[7] A. Papoulis, PROBABILITY, RANDOM VARIABLES, AND STOCHASTIC PROCESSES, McGraw-Hill, Inc., New york, 1965, Chapters 9 and 10.

(7) For ensemble averaging, average the corresponding frequency components for several input time histories.

(8) Perform adjacent frequency component averaging if required.

Each of these steps is discussed in the following paragraphs. The step sequence for the Cross-Spectral Density Computation is similar and is discussed subsequently.

Fast Fourier Transform (FFT) algorithms gain significant computational speed advantages when the number of input sample points to be processed is a power of two. If the number of input sample points to the Auto-Spectral Density algorithm is a power of two, then no adjustment is required. If not, then the smallest power of two that is greater than the input data record is computed. If this number is larger than the available storage in the program, then the input record length is truncated to the next lower power of two. If the larger power of two can be accommodated by available storage, then the input data record is extended by adding additional zero-valued samples to the end of the record to form an extended record. The original data record length and the new extended record length are recorded for later use in Steps 3 and 6. If the input record is truncated or not extended, then the data record length and the extended record length are the same.

The mean level for the data record is reset to zero to avoid distortions in a constant level that are introduced by a window function.

The concept of a window function was introduced in Section 6.1.3 for Amplitude Spectra. Four windows are provided for the Auto-Spectral Density calculation and the temporal representation of these windows is shown in Figure 64 where "T" is the input data record length. The Half Cosine Window is a new introduction that is not available for the Amplitude Spectrum analysis. The value for each window at a time of zero is one. The effects of each window can be estimated by observing the Fourier Transform of each window in Figure 65. The effect of a window application is to convolve the Fourier Transform of the window with the Fourier Transform of the time history to which it is applied.

Thus, the Rectangle Window (i.e., no applied window) has the best frequency resolution but also has large side lobes that decay as 1/F where F is absolute frequency. The Hanning

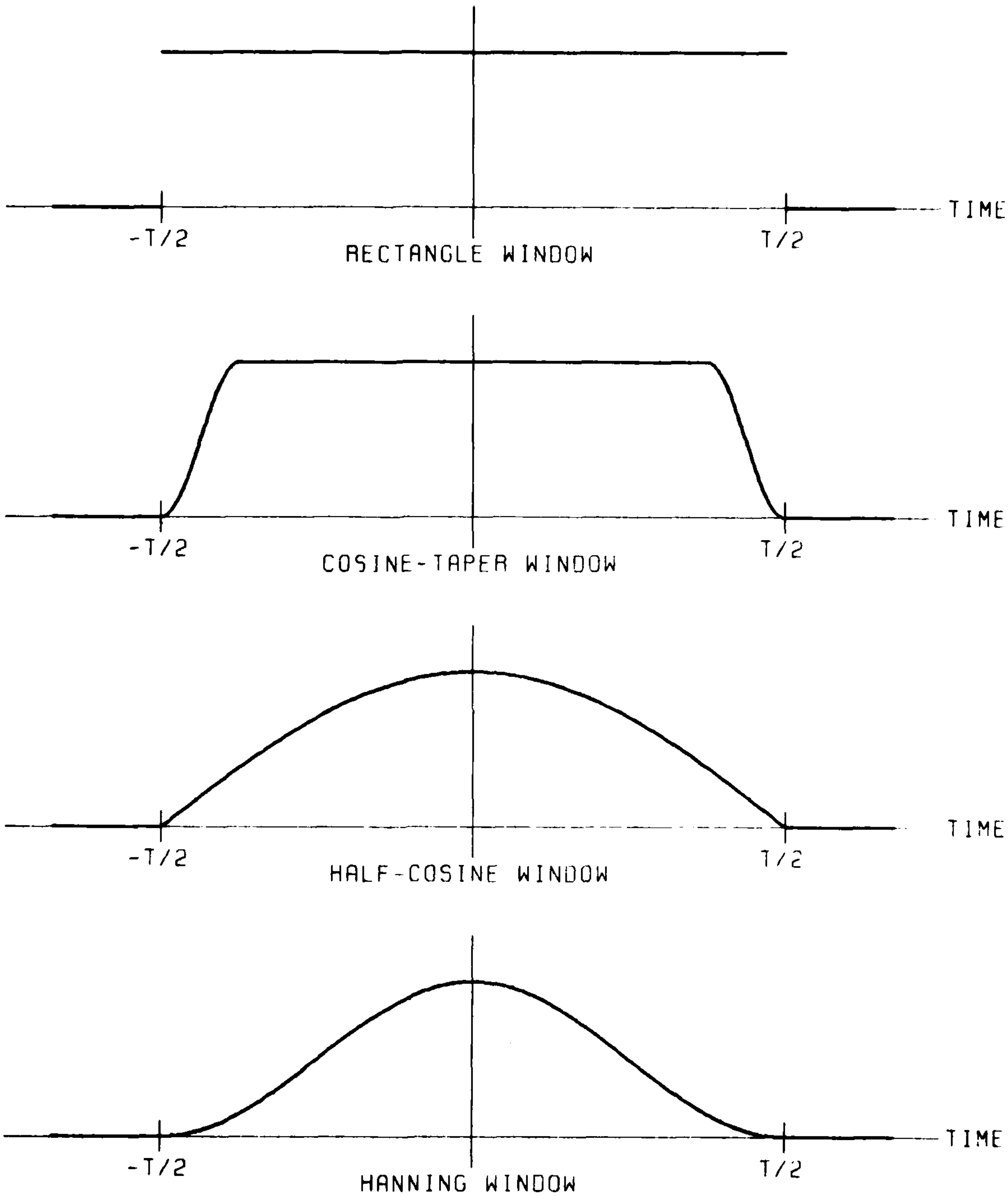


Figure 64. Temporal representation of window functions.

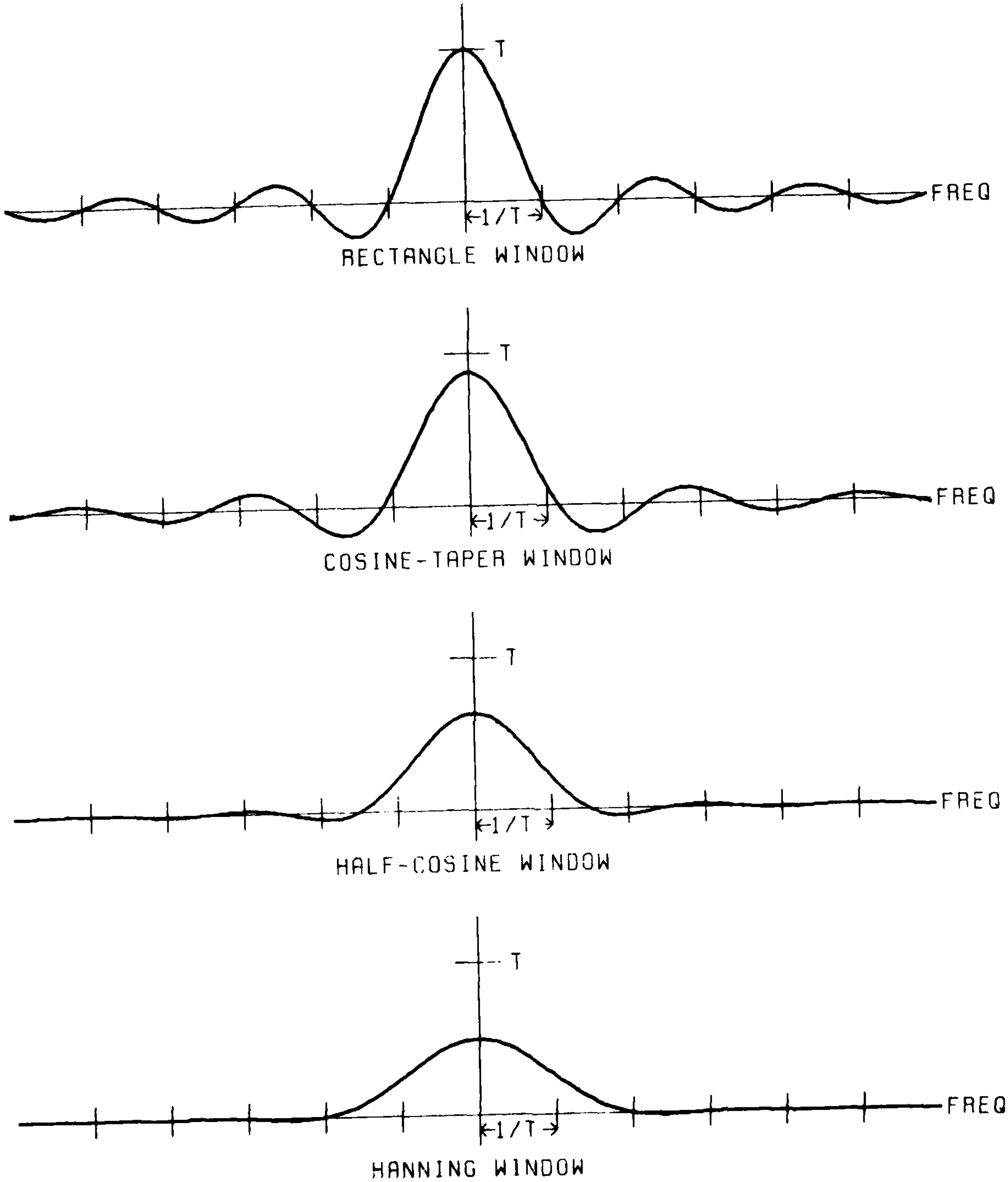


Figure 65. Frequency-domain representation of window functions.

Window has poor frequency resolution but the side lobes decay dramatically.[8] The Half Cosine Window is the default for this process and represnts a good compromise in window selection. This window exhibits excellent side-lobe decay and fair frequency resolution. However, the user is free to select the best window for his own particular application. The window selected is scaled in length and applied to the data record.

The FFT is then applied. The algorithm used calculates complex frequency components on the positive frequency axis. The zero-frequency component is discarded and the squared magnitude for each of the other complex components is calculated.

Multiplication of the squared magnitudes by a constant is required to arrive at a properly normalized Auto-Spectral Density. Several factors are required in this constant.

$$K = 2Q\Delta t/M$$

where K is the normalization constant, Q is a factor to compensate for the applied window, Δt is the sample interval in seconds, and M is the number of samples in the data record. Q is computed from

$$Q = \sum_{i=1}^{M} F_i^2 / \sum_{i=1}^{M} (W_i F_i)^2$$

where W_i is the window function value for data point i and F_i is the digitized input function value for data point i (recall that the mean value has been removed from the input data in step 2). The equation computes the change in variance to the input data from the application of the window.

The traditional definition of the Auto-Spectral Density estimator is (see Reference 7).

$$S_{xx}(\omega) \cong \frac{1}{T} \left| \int_{-T/2}^{T/2} x(t) e^{-2\pi i \omega t} dt \right|^2$$

[8]S. D. Stearns, DIGITAL SIGNAL ANALYSIS, Hayden Book Company, Inc., Rochelle Park, New Jersey, 1975, Chapter 14.

where

T = record length

x = input function

ω = frequency in Hz

i = $\sqrt{-1}$

S_{xx} = two-sided auto-spectral density estimate

so that a factor of $1/T = 1/M\Delta t$ is required. The FFT subroutine that is used is not scaled to any particular time interval. When the FFT output is considered to be an integration, as in the above equation, then the FFT output must be multiplied by the sampling interval, Δt. Since the squares of the FFT outputs are being processed, a factor of $(\Delta t)^2$ is required.

A factor of 2 is required because a single-sided spectrum is computed. That is, the integral of the computed Auto-Spectral Density from zero to the Nyquist Frequency for the sample rate should equal the variance for the input time history. Combining these factors K is computed:

$$K = 2(\Delta t)^2(Q)(1/M\Delta t) = 2Q\Delta t/M$$

as stated previously.

Steps (7) and (8) perform averaging to reduce the variance in the Auto-Spectral Density estimate for each frequency. Ensemble averaging, Step 7, takes the results of Steps 1 through 6 for more than one input record and averages the corresponding frequency components. Ensemble averaging provides excellent results without loss of frequency resolution, but this process requires more time for setup and computation. Adjacent point averaging, Step 8, takes the results of Steps 1 through 7 and averages sets of adjacent frequency components to compute a final estimate for the central component of each set. Adjacent point averaging reduces frequency resolution but does not require more time for setup and computation. Both adjacent point averaging and ensemble averaging may be used together.

The Cross-Spectral Density computation is very similar to the computation for Auto-Spectral Density. Pairs of input functions are processed in Steps 1 through 4. The squared magnitude computation of Step 5 is changed to complex multiplication of

each frequency component from the second input record of each pair by the complex conjugate of the corresponding frequency component from the first input record. Steps 6 through 8 are the same except that complex arithmetic is used. The final result is converted to magnitude and phase parts of the complex frequency components. Magnitude is the TOP double-row element, and phase in degrees is the BOTTOM double-row element. The Cross-Spectral Density output is again a single-sided spectrum because this function is even for real input records.

6.1.8 Frequency Response Analysis

Frequency Response analysis is used to determine the characteristics of linear systems. The Frequency Response of a linear system is the Fourier Transform of the impulse response of the system.[9] Alternatively, the Frequency Response is the complex change between each pair of corresponding input and output frequency components of a linear system. That is, if x(t) is the input and y(t) is the output of a linear system, and if x(t) and y(t) have Fourier Transforms X(w) and Y(w) respectively then

$$Y(w) = H(iw)\ X(w),$$

where H(iw) is the Frequency Response Function.

Nominally, one might expect that the Frequency Response could be computed from

$$H(iw) \stackrel{?}{=} Y(w)/X(w)$$

However, this method suffers from two major deficiencies. First, $X(w_o)$ may be zero from some value of w_o so that $H(iw_o)$ cannot be computed. Second, uncorrelated noise may be present in the measurements.

[9] R. Deutsch, SYSTEM ANALYSIS TECHNIQUES, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969, Chapter 5.

That is,

$$x(t) = a(t) + n(t)$$

$$y(t) = b(t) + m(t)$$

where x(t) and y(t) are the measured input and output signals, a(t) and b(t) are the true input and output signals, and n(t) and m(t) are uncorrelated noise signals. Ensemble averaging of the Fourier Transforms, X and Y, from different pairs of input and output records is not generally useful since the phase for corresponding frequency components of the member functions of a process is not constant. However, the phase difference between corresponding frequency components of a(t) and b(t) is constant for a linear system and this phase difference is computed by the Cross-Spectral Density. Thus, the ensemble-averaged Auto- and Cross-Spectral Density functions can be used to estimate the Frequency Response Function.

$$H(iw) \cong \frac{\overline{X^*(w)Y(w)}}{\overline{X^*(w)X(w)}}$$

where the overline represents ensemble averaging and the asterisk indicates the complex conjugate. The ensemble averaged Auto-Spectral Density, $\overline{X^*(w)X(w)}$, may still be zero or less than the computational noise generated in the process. Special consideration is given to this problem in the Frequency Response algorithm that is used.

For computational purposes, one or more pairs of input and corresponding output functions are processed. Steps 1 through 5, 7, and 9 from Paragraph 6.1.7 are performed to calculate auto-spectral density for the input records and cross-spectral density for the input and output records. Step 6 is not performed because the normalization factors will cancel when Step 9 is performed. Step 9 divides each Cross-Spectral Density component by the corresponding Auto-Spectral Density component. A minimum threshold value is set for the Auto-Spectral Density component based upon the maximum Auto-Spectral Density component that occurs. For Auto-Spectral Density components below the threshold, the Frequency Response output value is arbitrarily set to zero. All of the complex output values are converted to magnitude and phase as TOP and BOTTOM double-row elements, respectively, for display by the program.

6.1.9 Coherence Function Analysis

The Coherence Function is a useful tool for determining whether a set of pairs of measured signals represents the inputs to and corresponding outputs from a linear system. If it can be established that the system in question is, in fact, linear and that the only comtamination to the input and output measurements is uncorrelated noise, then the coherence function can be used to estimate the relative amount of noise present for all frequencies of interest.

The calculation of the Coherence Function, Q(w), is computed from:

$$Q(w) = \left| \overline{Y(w)X^*(w)} \right|^2 / \left(\overline{X(w)X^*(w)}\ \overline{Y(w)Y^*(w)} \right)$$

Notice that if the above equation is applied for a single pair of input records, x(t) and y(t), so that no ensemble averaging is performed, then the result is identically one for all frequencies. Thus, ensemble averaging is mandatory for a meaningful coherence function computation. The above equation can be broken into ensemble averaged Auto-Spectral Densities for the two sets of input functions, x(t) and y(t), and the ensemble averaged Cross-Spectral Density for both. These densities are computed as described in Paragraph 6.1.7 except that Step 6 is not performed. Then the densities are combined as indicated in the equation. If the divisor in the equation is zero for a particular frequency, the output value is arbitrarily set to zero for that frequency. All output for the Coherence Function is real and should lie between zero and one.

Values of the Coherence Function near one indicate high coherence or correlation for the corresponding x(t) and y(t) frequency components. Values near zero indicate low coherence for these components. When x(t) and y(t) represent measurements of the input and corresponding outputs of a linear system, the Coherence Function values are estimates of the relative amount of uncorrelated noise in these measurements. Reference 10 gives an extensive discussion of use of the Coherence Function in determining the amount of ensemble averaging required for accurate estimates of frequency response.

---

[10] J. S. Bendat, and A. G. Piersol, MEASUREMENT AND ANALYSIS OF RANDOM DATA, John Wiley & Sons, Inc., New York, 1966, Chapter 5.

6.1.10 Auto- and Cross-Correlation

The functions computed for the Auto- and Cross-Correlation analyses are actually auto- and cross-covariance. The only difference is that the mean values for input data records are reset to zero before processing. The definitions of auto-correlation, R_{xx}, and cross-correlation, R_{xy}, for real input processes x(t) and y(t) are (see Reference 7):

$$R_{xx}(\tau) = E\{x(t+\tau)x(t)\}$$

$$R_{yx}(\tau) = E\{x(t+\tau)y(t)\}$$

Where E{ } denotes expected value. For finite length, digitized input records, the auto-correlation can be estimated from

$$R_{xx}(j\Delta t) \cong \sum_{i=0}^{N-j-1} x((i+j)\Delta t)x(i\Delta t)/(N-j)$$

where Δt is the sample interval, $j\Delta t$ is the offset or lag ($j=0,1\ldots, N-1$), and N is the total number of samples in the record. Two formulas are required for a cross-correlation estimation

$$R_{yx}(j\Delta t) \cong \sum_{i=0}^{N-j-1} x((i+j)\Delta t)y(i\Delta t)/(N-j)$$

for $j=0, 1, 2\ldots, N-1$, and

$$R_{yx}(j\Delta t) \cong \sum_{i=-j}^{N-1} x((i+j)\Delta t)y(i\Delta t)/(N+j)$$

for $j= -N+1, -N+2, \ldots, -1$. Notice that auto-correlation is a special case of cross-correlation with $y(t)= x(t)$ for all t. In this case, $R_{xx}(-j\Delta t)=R_{xx}(j\Delta t)$, so that computation of auto-correlation for the non-negative offsets is sufficient.

The above equations are actually evaluated using the Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFT) algorithms as dictated by the Convolution Theoren (see Reference 7, Chapters 5 and 10). That is,

$$R_{xx}(i\Delta t) \overset{?}{\cong} IFT_i\{ FFT\{x(t)\}^2\}$$

$$R_{yx}(i\Delta t) \overset{?}{\cong} IFT_i\{FFT^*\{y(t)\}FFT\{x(t)\}\}$$

with divisors of N perhaps required depending upon the characteristics of the FFT subroutine that is used. If

$$FFT_j\{f\} = F_j = \frac{1}{N} \sum_{k=o}^{N-1} f(k\Delta t)e^{\frac{-2\pi ijk}{N}}$$

$$IFT_k\{F\} = f_k = \sum_{j=o}^{N-1} F_j e^{\frac{2\pi ijk}{N}}$$

where N is the number of values of the sampled function, f, then:

$$IFT_m\{FFT^*\{y\}FFT\{x\}\} = \sum_{j=o}^{N-1} X_j \, Y_j^* \, e^{\frac{2\pi ijm}{N}}$$

$$= \frac{1}{N} \sum_{j=o}^{N-1} \left[X_j \, e^{\frac{2\pi ijm}{N}} \sum_{k=o}^{N-1} y(k\Delta t)e^{\frac{2\pi ijk}{N}} \right]$$

$$= \frac{1}{N} \sum_{j=o}^{N-1} \sum_{k=o}^{N-1} y(k\Delta t)X_j \, e^{\frac{2\pi ij(k+m)}{N}}$$

$$= \frac{1}{N} \sum_{k=o}^{N-1} \left[y(k\Delta t) \sum_{j=o}^{N-1} X_j \, e^{\frac{2\pi ij(k+m)}{N}} \right]$$

Now assume that x and y are periodic functions with period $N\Delta t$. That is,

$$x((\ell+kN)\Delta t) = x(\ell\Delta t)$$

$$y((\ell+kN)\Delta t) = y(\ell\Delta t) \text{ for } k \text{ an integer}$$

Then from the above relationship

$$IFT_m\{FFT^*\{y\}FFT\{x\}\} = \frac{1}{N} \sum_{k=o}^{N-1} y(k\Delta t)x((k+M)\Delta t)$$

$$= \frac{1}{N} \sum_{k=o}^{N-m-1} y(k\Delta t)x((k+m)\Delta t) + \sum_{k=N-m}^{N-1} y(k\Delta t)x((k+m-N)\Delta t)$$

Clearly the above computation is not the same as the desired formula for R_{yx}, although the first term inside the brackets is very similar to the computation that is wanted. The second term inside the brackets includes cross-quantities that are valid only when the functions x and y are truely periodic with period $N\Delta t$. In fact, the above computational method is used to compute auto- and cross-correlation for cycle-averaged data. However, for non-periodic input functions, the second term in the brackets must be eliminated by the following method.

Consider two new functions, $\hat{x}$ and $\hat{y}$ such that

$$\hat{x}(i\Delta t) = \begin{cases} x(i\Delta t) & , i=0,1,\ldots,N-1 \\ 0 & , i=N,N+1,\ldots,M-1 \end{cases}$$

and

$$\hat{y}(i\Delta t) = \begin{cases} y(i\Delta t) & , i=0,1,\ldots,N-1 \\ 0 & , i=N,N+1,\ldots,M-1 \end{cases}$$

Also define

$$\hat{x}((i+kM)\Delta t) = \hat{x}(k\Delta t)$$

$$\hat{y}((i+kM)\Delta t) = \hat{y}(k\Delta t)$$

Now perform the FFT and IFT computation on the new functions, $\hat{x}$ and $\hat{y}$.

$$IFT_m\{FFT^*\{\hat{y}\}FFT\{\hat{x}\}\} =$$

$$\frac{1}{M}\left[\sum_{k=o}^{N-m-1} \hat{y}(k\Delta t)\hat{x}((k+m)\Delta t) + \sum_{k=N-m}^{M-1} \hat{y}(k\Delta t)\hat{x}((k+m)\Delta t)\right]$$

Since $\hat{x}((k+m)\Delta t) = 0$ for $k=N-m,\ldots,N-1$ and $\hat{y}(k\Delta t) = 0$ for $k=N,\ldots,M-1$ the second term in the brackets is zero. Also, for each index used in the first term, $\hat{y}=y$ and $\hat{x}=x$. This formulation assumes m is non-negative but the formulation for negative n is similar. Thus,

$$IFT_m\{FFT^*\{\hat{y}\}FFT\{\hat{x}\}\} = \frac{1}{M}\sum_{k=o}^{N-m-1} y(k\Delta t)x((k+m)\Delta t)$$

and

$$R_{yx}(n\Delta t) \cong \frac{M}{(N-m)} IFT_m\{FFT^*\{\hat{y}\}FFT\{\hat{x}\}\}$$

$$R_{xx}(m\Delta t) \cong \frac{M}{(N-m)} IFT_m\{FFT\{\hat{x}\}^2\}$$

16 F Hence, the steps for Auto-Correlation (auto-covariance) estimation are:

(1) Reset mean value for input record to zero.

(2) For non-cycle averaged data, extend the record to at least double the original length and to an exact power of two number of samples. Fill extended record with zeros.

(3) Perform FFT on extended record.

(4) Compute magnitude squared for each frequency component.

(5) Inverse FFT and discard negative offset values.

(6) Adjust for number of samples in the input record and the length of the extended record. For extended records, apply the N/(N-i) factor to each offset.

(7) If required, normalize the output value for each offset by dividing it by the zero offset value.

$\{\hat{R}_{xx}(\tau) = R_{xx}(\tau)/R_{xx}(0)\}$.

(8) Perform ensemble averaging of the output records, if required.

Notice that in Step 2 the length of a cycle-averaged record need not be adjusted to a power of two samples because the number of samples in a cycle-averaged record is already a power of two.

Normalization for the Auto-Correlation, Step 7, is an option that is provided to express auto-correlation in terms of the relative effects of offsets in the data. The zero offset value is set to one and all other output values should be between zero and one. However, the Auto-Correlation values that are computed are only estimates of the true auto-correlation of a process and the accuracy of these estimates decrease as the offsets increase.

The method for computation of Cross-Correlation is very similar to the method for Auto-Correlation except that two input records are processed. The order of these records is important in interpretation of the final output. A particular offset, τ, means that the first function is shifted forward in time (i.e., to the left on a typical Cartesian coordinate plot) by τ seconds with respect to the second input function. In Step 4, the product of the complex conjugate of each frequency component of the first record and the corresponding frequency component from the second record is computed. In Step 5, negative offset values are retained.

The method used for normalization of the Cross-Correlation in Step 7 is described in Reference 11.

That is,

$$\hat{R}_{xy}(\tau) = R_{xy}(\tau)/\sqrt{R_{xx}(0)\ R_{yy}(0)}$$

Computation of R_{xx} and R_{yy} to obtain the single values $R_{xx}(0)$ and $R_{yy}(0)$ is avoided. These values are identical to the mean squares of the input data records after the mean levels are reset to zero.

An implicit assumption has been made for the derivations in this paragraph that the input sample record or records represent stationary stochastic processes. That is,

---

[11] P. R. Roth, EFFECTIVE MEASUREMENTS USING DIGITAL SIGNAL ANALYSIS, IEEE Spectrum, April 1971, pp. 62-70.

$$R_{xx}(\tau) = E\{x(t+\tau)x(t)\}$$

is not a function of time, t, but only of time offset, τ. However, these analyses may also have useful application for processes that are not strictly stationary. Use of these methods for analysis of the input and output of linear systems is described in Reference 11.

6.1.11 Basic Statistical Computations

Three basic statistical computations are available in DATAMAP: sample mean, sample variance, and sample standard deviation. The user must assume the responsibility to ensure that a sufficient and representative sample of the population in question is provided. Input may include one or more time histories for the computation of one mean, variance, or standard deviation estimate (i.e., ensemble averaging is allowed for these estimates).

The sample mean is computed from

$$\bar{y} = \sum_{i=1}^{N} y_i/N$$

where y_i are the individual data samples and N is the total number of samples in all time histories that are input.

The sample variance is computed from

$$\sigma^2 = \sum_{i=1}^{N} \left[y_i^2 - N\bar{y}^2 \right] /(N-1)$$

This form is used because the sample mean is not known until the last time history is processed during ensemble averaging. The equation is derived from

$$\sigma^2 = \sum_{i=1}^{N} (y_i-\bar{y})^2/(N-1)$$

As indicated in Reference 12, the above equation uses the N-1 divisor instead of N to allow for the bias introduced by the sample mean, $\bar{y}$, instead of the true population mean in the estimate.

The sample standard deviation is computed as the square root of σ^2.

6.1.12 Normal Distribution Test

Frequently, it is useful to determine whether the sample points from one or more input records are distributed normally. In particular, a normal or Gaussian distribution is completely defined by the mean and variance of the data. The probability density function, p(y), for a normal distribution is

$$p(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(y-a)^2}{2\sigma^2}}$$

where σ^2 is the variance, a is the mean, and y is the random variable for the distribution (see Reference 12).

If it is desired to test the hypothesis that a population is normally distributed, a Chi-Square test can be performed. The Chi-Square test is applicable for other distributions, but in DATAMAP the test is limited to normal distributions. First, the mean and variance of the first input time history are established. Based upon these values, the random variable range is broken into segments or "bins." The size and location of these bins is computed so that the probability that any one sample will fall into any bin is the same for all the bins. The bins are positioned contiguously so that a sample must fall into one of the bins. The number of bins is selectable by the user, although a default of seven bins is specified. Qualitatively, if the auto-correlation values for the input data are low except near the zero-offset value, then the number of bins may be as many as 1/20th the total number of samples input. For more slowly varying time histories, more samples are required for each bin.

---

[12]B. Carnahan, H. A. Luther, and J. O. Wilkes, APPLIED NUMERICAL METHODS, John Wiley & Sons, Inc., New York, 1969, Chapter 8.

The steps followed in computation of a Chi-Squared value for input sample data are:

(1) Compute mean and variance for the first input sample record.

(2) Compute bin boundaries so that samples will fall evenly in the bins assuming the population is normally distributed.

(3) Distribute the input samples from the first record into the bins.

(4) Repeat Step 3 for all input records. Maintain a running computation of cumulative mean and variance for all input records.

(5) After all sample records have been processed, recompute the expected population of each bin based upon the new values of mean and variance for all sample records.

(6) If the expected content of any bin is now less than 4.5, notify the user of an error condition.

(7) Compute Chi-square from the expected and actual bin contents.

Steps 2 and 5 require use of the normal probability distribution function, which is generally computed from the error function, ERF. To avoid dependence on the IBM error function routine, the normal distribution function is computed by interpolation of a table.

The Chi-squared value is computed using the formula

$$\chi^2 = \sum_{i=1}^{N} \frac{(\left|X_i - E_i\right| - .5)^2}{E_i}$$

where

N = total number of bins

X_i = number of samples that fall in the ith bin

E_i = expected number of samples for the ith bin based upon an assumed normal distribution.

χ^2 = the Chi-Squared value

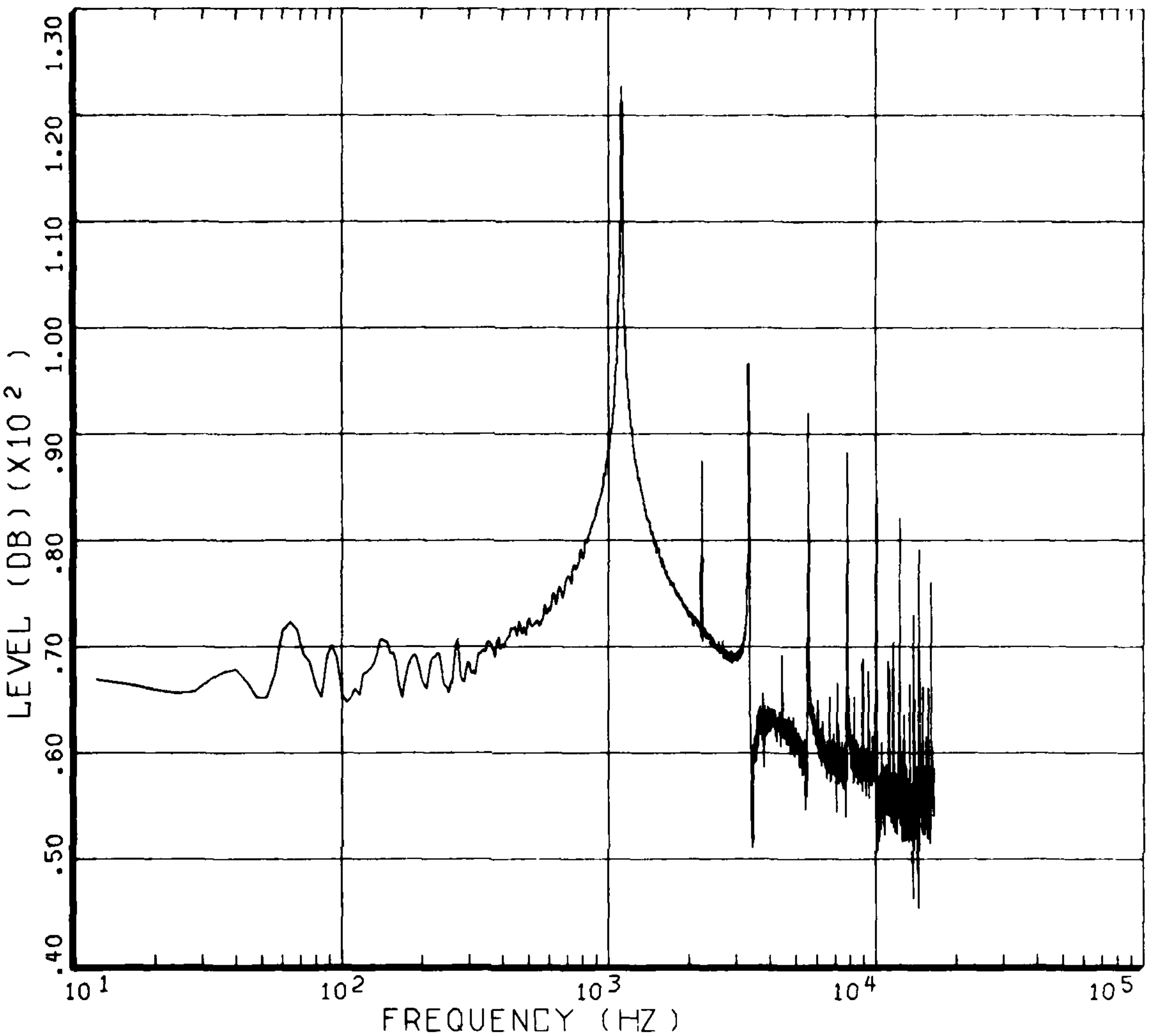
The .5 term is "Yate's correction" and is described in Reference 12.

The user must consider the number of degrees of freedom in the test in order to properly interpret the Chi-Squared result. This number is one less than the number of bins used in the calculation. Given the Chi-Squared value and the number of degrees of freedom, the user can consult Reference 10 or 12 or most mathematical tables to obtain a probability, P. If the population that is being sampled were truely normally distributed with the computed mean and variance, P is the probability that Chi-Squared could be at least as high as calculated for one experiement as performed in the above calculation. Thus, this test does not "prove" a normal distribution but should show inconsistency for distributions that deviate from normal.

6.1.13 Acoustic Analyses - General Comments

All of the acoustic analyses implemented in DATAMAP use the same basic approach to the processing of acoustic input data in the frequency domain. That is, input data are Fast Fourier Transformed (FFT'd) and then the individual Fourier components are operated upon as necessary to emulate some analog process. In most cases, this computational method is considerably faster than the direct emulation of analog methods using various digital filters. However, the raw FFT method used can allow frequency "leakage" as described in Paragraphs 6.1.3 and 6.1.7 because no window is applied before the FFT is used (i.e., the rectangle window is used). The effects of this leakage are apparent when a pure tone signal is processed (see Figure 66). Very high audio levels in one frequency band could affect the calculations for much lower levels in an adjacent frequency band. However, under most circumstances, noise is distributed much more smoothly in frequency. For example, Figure 22 shows a narrow band analysis of helicopter cabin noise. Thus, the acoustic analyses in DATAMAP will provide good estimates of various noise levels, but they will not satisfy rigorous requirements such as for the Federal Aviation Regulations, Part 36. Proposed future modifications to these analyses may remedy this technical deficiency.

All of the acoustic analyses assume that the input signal is scaled in units of ratio of pressure to the standard threshold



CALIBRATION TONE INPUT

8 HZ NARR BAND ANAL 206B CABIN NOISE

COUNTER 1 GROSS WT SHIP MODEL 206B
LONG CG SHIP ID

1/R849

BHT.USARTL DATAMAP (VERS 3.00 - 03/29/80) 04/19/80

Figure 66. Narrow-band analysis of calibration tone signal.

pressure of 2×10^{-5} Newtons per square meter. However, a calibration constant may always be specified in decibels for level adjustment.

6.1.14 Acoustic Narrow Band Analysis

Narrow Band Analysis is an emulation of an analog process that uses a constant width narrow band filter with an adjustable center frequency to sweep over the frequency domain of an acoustic signal. The acoustic signal is constantly replayed and the center frequency is swept at a constant rate from the minimum to the maximum desired frequency. A mean square computation is performed by a circuit that receives the filter output as input. Finally, the mean square output is converted to decibel units.

The digital narrow band filtering process is accomplished by weighting the frequency components of the Fourier Transformed signal in proportion to the transfer function of the analog filter that is being emulated. This filter is assumed to have a parabolic transfer function shape with a peak value at the selected center frequency. At the cutoff frequencies, which are separated by the selected frequency bandwidth and equidistant from the center frequency, the parabolic weighting function is 3 dB below the center frequency value of one. The skirts of the parabola are truncated when the weighting function is 60 dB down from the center frequency. That is, weights that are more than 60 dB down are arbitrarily set to zero. For computational reasons, the width of the filter may be adjusted to be an integer multiple of the reciprocal of the record length, 1/T. To assure a reasonably low variance in the output result, the bandwidth should be at least 2/T. The default bandwidth has been set to 8 Hz because the maximum number of points that can be processed in DATAMAP is 8192 and for a typical sample rate of 32767, the corresponding record length is .25 second.

The steps used for the Narrow Band Analysis are:

(1) FFT the acoustic input signal.

(2) Compute the magnitude squared of the frequency components.

(3) Given the difference in frequency between the filter center frequency and the 60 dB down point for the filter, find the lowest frequency component that is at least as high as this difference. This is the first center frequency.

(4) For each center frequency, weight the magnitude square of the frequency co-efficients by the magnitude square of the centered filter frequency response function.

(5) For each center frequency, form the sum of the weighted co-efficients from step 4.

(6) Convert to decibel units by multiplying the logarithim of each sum from step 5 by 10.

(7) For each center frequency value, add the calibration level specified.

Steps (4) and (5) are actually computed with a recursive algorithm. Define

f_c = center frequency for a band

Δf = frequency resolution for the FFT, which is the reciprocal of the input record length

f_o = one-half the specified band width

i = a positive integer

p,r = positive real numbers.

Y_i = magnitude squared of FFT output for frequency $(i-1)\Delta f$

Then set

$$(p-1)\Delta f = f_c$$

$$r\Delta f = f_o$$

The desired parabolic filter response function is

$$H^2((i-1)\Delta f) = H_i^2 = \frac{(10^{-3/10}-1)}{r^2}(i-p)^2 + 1$$

To assure that the first band is centered at the lowest possible frequency, while keeping the non-zero skirts of the filter response function on the positive frequency axis, set $H_1^2 = 10^{-6}$ for the first band. Then solving for p

$$p = 1 + r\sqrt{\frac{1-10^{-6}}{1-10^{-3/10}}} = 1 + r(1.4158945)$$

The variable "r" (i.e., $f_o/\Delta f$) may be adjusted as necessary so that "p" is an integer. Since the parabola is symetrical about the center frequency, the filter response for the first band is again 10^{-6} at the frequency $(j-1)\Delta f$ where

$$j = 2p-1$$

Thus, the mean square output from the first band is

$$S_1 = Y_1H_1^2 + Y_2H_2^2 + \ldots + Y_jH_j^2 ,$$

where Y_i is the amplitude square of the Fourier coefficients.

In general, for the n'th band for center frequency $\frac{(2n+j-2)}{2}\Delta f$, the mean square output is

$$S_n = \sum_{i=1}^{j} Y_{i+n-1}\, H_i^2 = \sum_{m=n}^{j+n-1} Y_m\, H_{m-n+1}^2 .$$

Now define

$$b = \frac{10^{-3/10}-1}{r^2}$$

so that

$$H_i^2 = \frac{10^{-3/10}-1}{r^2}(i-p)^2 + 1 = b(i-p)^2 + 1$$

Substitution and expansion will show

$$H_{m-n+2}^2 = H_{m-n}^2 + 4b(m-n-p) + 4b$$

$$2S_n - S_{n-1} = \sum_{m=n}^{j+n-1} Y_m(H_{m-n}^2 - 2b) - Y_{n-1}(H_{-1}^2 - 4b(1+p) + 4b)$$

$$+ Y_{j+n-1}(H_{j-1}^2 + 4b(j-1-p) + 4b)$$

setting

$$a_1 = H_{-1}^2 - 4b(1+p) + 4b$$

$$a_2 = H_o^2 - 2b$$

$$a_3 = H_{j-1}^2 + 4b(j-1-p) + 4b$$

$$a_4 = H_j^2 - 2b$$

Then

$$S_{n+1} = 2S_n - S_{n-1} + a_1 Y_{n-1} - a_2 Y_n - a_3 Y_{j+n-1} + a_4 Y_{j+n}$$

$$+ 2b \sum_{m=n+1}^{j+n} Y_m$$

Set

$$U_k = \sum_{m=k}^{j+k-1} Y_m$$

and initially compute

$$S_1 = b \sum_{i=1}^{j} Y_i(i-p)^2 + U_1$$

$$S_2 = b \sum_{i=1}^{j} Y_{i+1}(i-p)^2 + U_2$$

Then for subsequent bands

$$U_{n+1} = U_n + Y_{j+n} - Y_n$$

$$S_{n+1} = 2S_n - S_{n-1} + a_1Y_{n-1} - a_2Y_n - a_3Y_{j+n-1} + a_4Y_{j+n}$$
$$+ 2bU_{n+1}$$

This calculation is used for steps (4) and (5) in the overall narrow band analysis computation algorithm.

6.1.15 Octave and Third Octave Analyses

The Octave and Third Octave analyses are quite similar to the Narrow Band Analysis. That is, the analyses emulate the application of band pass filters to the input data and the computation of the mean square output of each filter. However, for Octave and Third-Octave analysis, these filters are predefined in location and width according to Table 11 for Octaves and Table 12 for Third-Octaves (some rounding is performed on the Table values). These filters do not have a constant width as do the Narrow-Band analysis filters. Instead, the filter widths are a constant percentage of the center frequency for each filter.

The weighting function applied to emulate each constant percentage filter is flat near the center frequency with a constant value of one. At 73.5 and 133.4 percent of each center frequency, the octave filter weighting functions break downward so that at the cutoff frequencies they are 3 dB down. At 35.9 and 240 percent of the center frequency, the weighting functions are 40 dB down. Weighting function values that are further down than 40 dB are reset to zero.

The third octave weighting functions break downward at 91.1 and 107.6 percent of the center frequency. The functions are 3dB down at the cutoff frequencies and 60 dB down at one-half and twice the center frequencies. Weighting function values that are further down than 60 dB are reset to zero.

Application of the Octave and Third-Octave analysis follows the same sequence of steps as the Narrow-Band analysis as described in Paragraph 6.1.12. However, in Step 3 the lowest octave or third-octave that can be computed is based upon a

TABLE 11. OCTAVE FILTER DEFINITION

Octave Number	Lower Cutoff Frequency (Hz)	Center Frequency (Hz)	Upper Cutoff Frequency (Hz)
1	11	16	22
2	22	31.5	45
3	45	63	89
4	89	125	177
5	177	250	354
6	354	500	707
7	707	1000	1414
8	1414	2000	2828
9	2828	4000	5656
10	5656	8000	11312
11	11312	16000	22625

TABLE 12. THIRD OCTAVE FILTER DEFINITION

Third Octave Number	Lower Cutoff Frequency (Hz)	Center Frequency (Hz)	Upper Cutoff Frequency (Hz)
1	11	12.5	14
2	14	16	18
3	18	20	22
4	22	25	28
5	28	31.5	35
6	36	40	45
7	45	50	56
8	56	63	71
9	71	80	90
10	90	100	112
11	112	125	140
12	140	160	180
13	180	200	225
14	225	250	281
15	281	315	354
16	354	400	449
17	449	500	561
18	561	630	708
19	708	800	898
20	898	1000	1122
21	1122	1250	1403
22	1403	1600	1796
23	1796	2000	2245
24	2245	2500	2806
25	2806	3150	3536
26	3536	4000	4490
27	4490	5000	5612
28	5612	6300	7080
29	7080	8000	8980
30	8980	10000	11225
31	11225	12500	14031
32	14031	16000	17960

comparison of the frequency resolution (i.e., the reciprocal of the record length) and the width of the octave or third-octave (i.e., the difference between the upper and lower cut-off frequencies). Octave or Third-Octave values are not computed for octaves or third-octaves that are narrower than the frequency resolution. For example, for a record length of .25 second the frequency resolution is four Hz. The width of the first third octave is about three Hz so no value is computed for this third octave.

The highest octave or third octave that can be computed is determined by the Nyquist Frequency for the data (i.e., one-half the sample rate). The upper cutoff frequency for the highest octave or third octave that is computed must be at or below the Nyquist Frequency. However, the octaves and third-octaves that can be computed are restricted to those listed in Tables 11 and 12.

Following is a description of the method of computation of the weighting functions for the octave filters. Let f_c be the center frequency for a filter. Then, the filter response function can be described as G(f) (in decibels) where

$$G(f) = \begin{cases} -\infty & \text{if } f < .359\, f_c \\ G_\ell(f) & \text{if } .359\, f_c \le f < .735\, f_c \\ 0 & \text{if } .735\, f_c \le f \le 1.334\, f_c \\ G_r(f) & \text{if } 1.334\, f_c < f \le 2.40\, f_c \\ -\infty & \text{if } 2.40\, f_c < f \end{cases}$$

where

$$G_\ell(f) = \frac{44\sqrt{2}}{f_c(2-\sqrt{2})}(f - f_c/2) - 25 \text{ (in dB)}$$

$$G_r(f) = \frac{22}{f_c(\sqrt{2}-2)}(f - 2f_c) - 25 \text{(in dB)}$$

These equations can be restated

$$G_\ell(f) = g_1 f + g_2$$

$$G_r(f) = h_1 f + h_2$$

Now filter outputs will be computed by summation in the pressure ratio range using

$$S = \frac{1}{2} \sum_{i=i_1}^{i_2} Y_i \; H_i^2$$

where $G(i\Delta f) \geq -40$ for $i_1 \leq i \leq i_2$ and

$$H_i = H(i\Delta f) = H(f) = 10^{\frac{G(f)}{20}}$$

For the frequency ranges where G_ℓ and G_r are used

$$H_\ell(i\Delta f) = 10^{\frac{g_1}{20} i\Delta f + \frac{g_2}{20}}$$

$$H_r(i\Delta f) = 10^{\frac{h_1}{20} i\Delta f + \frac{h_2}{20}}$$

let

$$g = 10^{\frac{g_1 \Delta f}{20}}$$

$$h = 10^{\frac{h_1 \Delta f}{20}}$$

then

$$H_\ell((i+1)\Delta f) = g \; H_\ell(i\Delta f)$$

$$H_r((i+1)\Delta f) = h \; H_\ell(i\Delta f)$$

Thus, the weighting function values are computed recursively in the non-flat portion of the frequency band. The above analysis was performed for a single octave filter weighting function. However, the analysis is general so that the computational method can be applied for each octave center frequency.

The computational method for the third octaves is the same as the recursive method for the full octaves, and the analysis is the same except that the constants are modified.

6.1.16 Perceived Noise Level Analysis

Computation of Perceived Noise Level is based upon a standard "perceived noisiness" function that is expressed in units of "Noys." The perceived noisiness function has two independent variables, frequency and sound pressure level in decibels. Following are the steps required for the Perceived Noise Level computation.

(1) Compute third-octave levels as described in paragraph 6.1.14.

(2) Compute the equivalent perceived noisiness, N_i, for each third octave from the sound pressure level and the center frequency.

(3) Determine the third octave, K, that has the largest perceived noisiness level and that lies between 7 and 30 inclusive.

(4) Combine the perceived noisiness levels according to the formula.

17 F

$$N_T = N_k + .15 \left(\left(\sum_{i=7}^{30} N_i \right) - N_k \right)$$

$$= .85N_k + .15 \left(\sum_{i=7}^{30} N_i \right)$$

where

N_i = the perceived noisiness levels

i = third octave numbers from Table 12

k = the noisiest third octave

(5) Convert the resultant combined noisiness level to sound pressure level in decibels assuming a frequency of 1000 Hz.

The perceived noisiness values are derived from a formula specified in Reference 13. However, as stated in Paragraph 6.1.11, the overall method used is not in technical conformity to this reference.

The output of this analysis is a single value in units of PNdB.

6.1.17 Acoustic Weighting Networks

The acoustic weighting networks (A, B, C, and D) are applied as weighting functions similar to the weighting functions for the Narrow Band, Octave, and Third-Octave analyses. However, for each weighting network applied, a single weighting function is used that spans the audible frequency range from about 10 Hz to about 16,000 Hz. The A, B, and C weighting networks are taken from Reference 14. No standard is available for the D network but the functional values were taken from the nominal values for a sound level meter.[15]

The A, B, C, and D weighting functions are implemented in the program as a set of five analytic functions for each weighting network. Each analytic function covers a particular frequency band and closely approximates the prescribed weighting network values. Thus, a weighting function value can be computed for each frequency component. A single sound pressure level is calculated for an input record and a specified weighting network from the formulas 17
B

$$Z_k = \frac{1}{2} \sum_{i=1}^{N} F_i^2 H_{i,k}^2$$

[13]Federal Aviation Regulations, Part 36, Appendix B, Department of Transportation, Federal Aviation Administration, April 1978.

[14]RECOMMENDATIONS FOR SOUND LEVEL METERS, International Electrotechnical Commission, Geneva, Switzerland, 1961.

[15]Instructions and Applications, IMPULSE PRECISION SOUND LEVEL METER TYPE 2204, Brüel & Kjaer, Copenhagen, Denmark, July 1969.

$$dBk = 10 \log_{10}(Z_k) + C$$

where

F_i = i'th frequency component of input record

$H_{i,k}$ = i'th frequency weighting function (transfer function) for weighting network k

k = weighting network (A, B, C, or D)

N = Number of computed frequency components from Fast Fourier Transform

C = Attenuation correction factor in dB

6.2 DERIVATIONS

6.2.1 Rotor Azimuth

Although rotor azimuth is a measured or directly simulated parameter, the rotor azimuth data stream must be processed to yield useable azimuth angles. The input data stream may consist of azimuth values with a range from 0.0 through 360.0 degrees or it may be a sequence of negative values broken by a single or two adjacent positive values that occur nominally at a zero degree azimuth position (red blade over the tail-boom for OLS data). For the pulse encoded input, a time instant is found for each positive value or adjacent pair of positive values. For the input in degrees, a time instant is found for each zero degree value. The cummulative correction offset angle formed from the initial group and geometric group offset angles (see Section 5.8) is then applied by interpolating between adjacent nominal zero degree time instants to get corrected zero degree time instants. For pulse encoded input data, jitter in the time instants is minimized by averaging each set of three zero degree time instants. For degree encoded input data, no jitter correction is performed. The azimuth angle associated with each time instant required for processing or for output is then computed from estimated time instants of 360K degrees (K an integer) using linear interpolation.

6.2.2 Vehicle True Airspeeds

For measured indicated airspeed input in knots squared, several computations are required to determine vehicle true airspeed. The parameters, Outside Air Temperature (°C), Boom

System Static Pressure (psia), and Boom System Airspeed (knots squared), are recorded on tape and are used in the computation.

First, the Boom System Airspeed in knots squared is smoothed by averaging all values from three adjacent rotor cycles, and one smoothed value is produced for each rotor cycle. Then, from the smoothed Boom System Airspeed, $v_i^2(t)$, calibrated airspeed is determined according to

$$v_c(t) = mv_i(t) + b,$$

where the constants m and b are not the calibration constants on tape and depend instead on the particular gage that was used and on the airspeed range of interest. Vehicle true airspeed, $v_T(t)$, is then computed from

$$V_T(t) = \frac{v_c(t)}{\sqrt{\sigma(t)}}, \text{ knots}$$

where

$$\sigma(t) = \frac{\rho(t)}{\rho_o},$$

ρ_o is the sea level standard air density (.002378 slug/ft^3) and

$$\rho(t) = \frac{\rho_o \, T_o \, P_s(t)}{P_o \, T_A(t)}, \ \frac{\text{slug}}{\text{ft}^3}.$$

$P_s(t)$ is the Boom System Static Pressure, T_o is the sea level standard temperature (518.7°R), P_o is the sea level standard pressure (14.7 psi), and $T_A(t)$ is the outside air temperature in °R, which is computed from measurements by

$$T_A(t) = 1.8 \, T(t) + 491.7,$$

where T(t) is the measured outside air temperature in °C.

For simulated or measured true airspeed input in knots, no further computations are required except to average the true airspeed values for three adjacent rotor cyles to obtain the true airspeed for the central cycle.

6.2.3 Rotor RPM

Rotor speed is evaluated for each zero degree azimuth instant by averaging the cycle intervals before and after that instant, taking the reciprocal of the average, and converting the resultant rotor frequency to RPM.

In order to obtain a function (i.e., RPM) with sample rate identical to other parameters of interest, a simplified cubic spline interpolation procedure is employed. The method seeks to minimize large oscillations between points and has the advantage that continuity of slope is achieved at measured data values. The method generally achieves smooth curves and is computationally efficient.

6.2.4 Rotor Mast Horsepower

Main or tail rotor mast horsepower is computed from the measured parameter, main rotor mast torque (in.-lb) or tail rotor mast torque (in.-lb), making use of the derived parameters, main rotor RPM, and tail rotor RPM. Mast horsepower for either main or tail rotor is then given by

$$HP(t) = K_0 K_1 K_2 \, Q(t) \, R_{PM}(t) \text{ , where}$$

K_0 is the conversion factor $2\pi/60$ converting the appropriate RPM to radians/sec, K_1 is the conversion factor 1 horsepower/550 $\frac{\text{ft-lb}}{\text{sec}}$, K_2 is the conversion from inches to feet, 1/12, $Q(t)$ is the appropriate mast torque, and $R_{PM}(t)$ is rotor RPM.

6.2.5 Thrust Coefficient

The thrust coefficient, C_T, is computed from Boom System static pressure (psia), outside air temperature (°C), and rotor RPM. C_T is given by

$$C_T = \frac{W}{\sigma \rho_o A (\Omega R)^2} \text{ ,}$$

where W is the ship gross weight or antitorque force, σ is the density ratio as described in Paragraph 6.2.2, ρ_o is sea level standard air density, A is area of the rotor disc (πR^2), and ΩR is the rotor tip speed

$\Omega R = R_{PM}(t)RK$, where

$R_{PM}(t)$ is rotor RPM, R is rotor radius, and K is the conversion from RPM to radians/sec ($2\pi/60$).

6.2.6 Torque Coefficient

The torque coefficient, C_Q, is computed from measurements of rotor mast torque (in.-lb), Boom System static pressure (psia), outside air temperature (°C), and the derived rotor RPM. In particular,

$$C_Q = \frac{Q(t)\ R_{PM}(t)\ K_0 K_1}{\sigma \rho_o\ A(\Omega)^3}\text{, where}$$

$Q(t)$ is rotor mast torque, $R_{PM}(t)$ is rotor RPM, K_0 is conversion from RPM to rad/sec ($2\pi/60$), K_1 is conversion from inches to feet (1/12), σ is air density ratio described in Paragraph 6.2.2, ρ_o is sea level standard air density (.002378 slug/ft^3), A is the area of the rotor disc (πR^2), and ΩR is the rotor tip speed described in Paragraph 6.2.5.

6.2.7 Blade Local Flow Magnitude and Direction

Blade local flow is derived from the two perpendicular measurements of differential pressure, q_1 and q_2, recorded from a single boundary layer button (BLB). In addition, boom system static pressure and outside air temperature measurements are used.

As described in Reference 1, a BLB consists of two total pressure tubes and a static port. These two tubes are mounted at an angle of 90 degrees to each other. For the OLS measurement program, the bisector of this right angle was always aligned with the blade chordline. However, in case the BLBs are rotated for another measurement program, the Processing Program allows the user to enter the angle, B, between the inboard pointing tube and the chordline. The default value for B is 45 degrees. The differential pressure measured using the inboard pointing tube and the static port is called $q_1(t)$ and the other differential pressure measurement is called $q_2(t)$.

The BLB flow angle, A_B, is the angle made by the flow and the bisector of the angle between the tubes. Negative $A_B(t)$ values imply flow more nearly aligned with the inboard pointing tube than the outboard pointing tube. $A_B(t)$ is computed by reference to calibration values that relate the flow angle in degrees and the ratio of the two differential pressures. The Info File format accommodates two calibration constants, C_1 and C_2, for each BLB (one number for each item code). The computational algorithm uses these two numbers as corrective multipliers where C_1 follows the inboard pointing item code in the Info File (the first item code listed for each sensor is the inboard) and C_2 follows the outboard pointing item code. C_1 applies to inboard or negative angles and C_2, to outboard or positive angles.

The calculation of BLB flow angle is made by one of two formulas

$$A_B(t) = C_1 F_N(q_2(t)/q_1(t))$$

if $q_2(t)$ is less than $q_1(t)$ (negative angle) or

$$A_B(t) = C_2 F_N(q_1(t)/q_2(t))$$

if $q_1(t)$ less than $q_2(t)$ (positive angle). F_N is one of two polynomial functions

$$F_1(R) = ((1.3R-11.3)R + 25.6)R - 15.6$$

$$F_2(R) = ((.059233R + 8.732)R - 26.23)R + 17.437$$

where $R = q_1/q_2$ or $R = q_2/q_1$. F_1 and F_2 are rough functional approximations of the BLB calibrations but are insufficiently accurate for application to individual BLB's without the correction factors C_1 or C_2.

F_1 is selected for negative angles when C_1 is positive and for positive angles when C_2 is negative. F_2 is selected for positive angles when C_2 is positive and for negative angles when C_1 is negative. Thus the 'fit' to the calibration points is made by selection of one of two third-degree functions and a correcting multiplicative factor based first on the relative size of q_1 and q_2 and then on the sign of the correction factor.

Figures 67 and 68 show representative comparisons of corrected polynomials (continuous lines) and calibration points (squares) for two BLBs. Similar corrected polynomial plots are available for every BLB used in the OLS program. In Figure 67, the function F_2 is used for both positive and negative flow angles. In Figure 68, the function F_2 is used for negative flow angles and F_1 is used for positive angles. Notice that no calibration values are available for angles greater than 25 degrees or less than -25 degrees. The Processing Program applies the same restriction to A_B so that A_B values greater than 25 degrees are reset to 25 degrees and A_B values less than -25 degrees are reset to -25 degrees.

Once the BLB flow angle is calculated, the velocity magnitude derivation can begin. The algorithm first determines the differential pressure, P(t), which should be measured by a tube pointed directly into the flow. The tube most nearly pointed into the flow is selected and the absolute angle between the flow direction and the tube direction is

$$U(t) = 45 - A_B(t)$$

The conversion function F_P is approximately

$$F_P(U(t)) \cong \frac{q(t)}{P(t)}$$

Where q(t) is $q_1(t)$ or $q_2(t)$ as selected and P(t) is the calculated pressure from direct flow into a tube at the same velocity magnitude. Preliminary calibration efforts showed that F_P does not change significantly with velocity magnitude. F_P is a third degree functional approximation of calibration points from all BLB's used in the OLS measurement program.

$$F_P(U(t)) = ((.0000052(U(t) - 20) - .0007158)(U(t) - 20) - .0046202)(U(t) - 20) + 1.003479$$

As shown in Figure 69, considerable variation is apparent in the calibration points. However, velocity is proportional to the square root of pressure. Thus, a worst case excursion of about 13 percent in pressure ratio becomes about 7 percent excursion in velocity.

Given a pressure measurement from one sensor, q(t), the angle made by the sensor tube with direct flow, U(t), the conversion function, F_P and the air density, as described in Paragraph 6.2.2, $\rho(t)$, the calculation of flow magnitude is straightforward. Since the direct flow pressure is

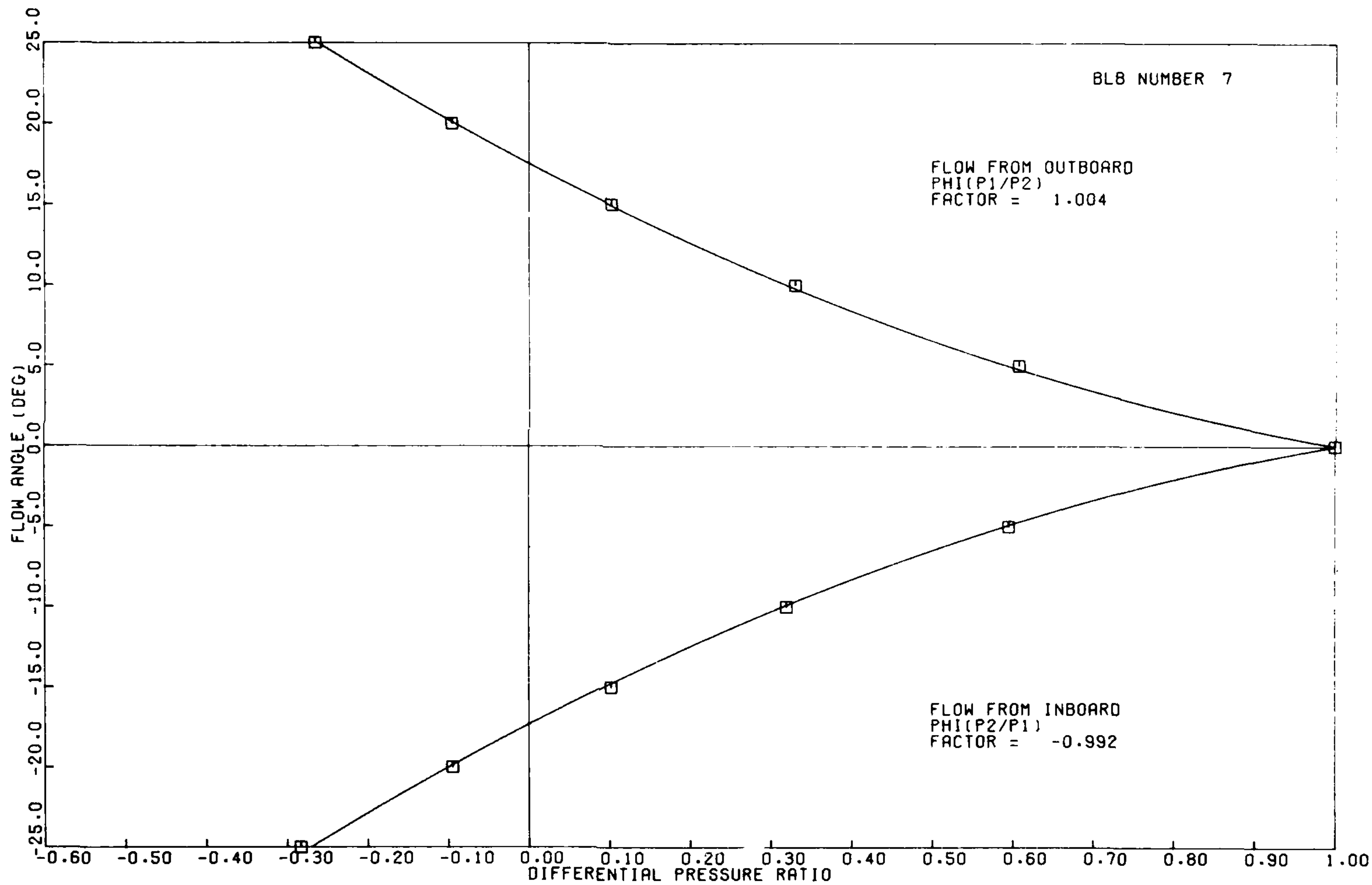


Figure 67. Calibration of BLB number 7.

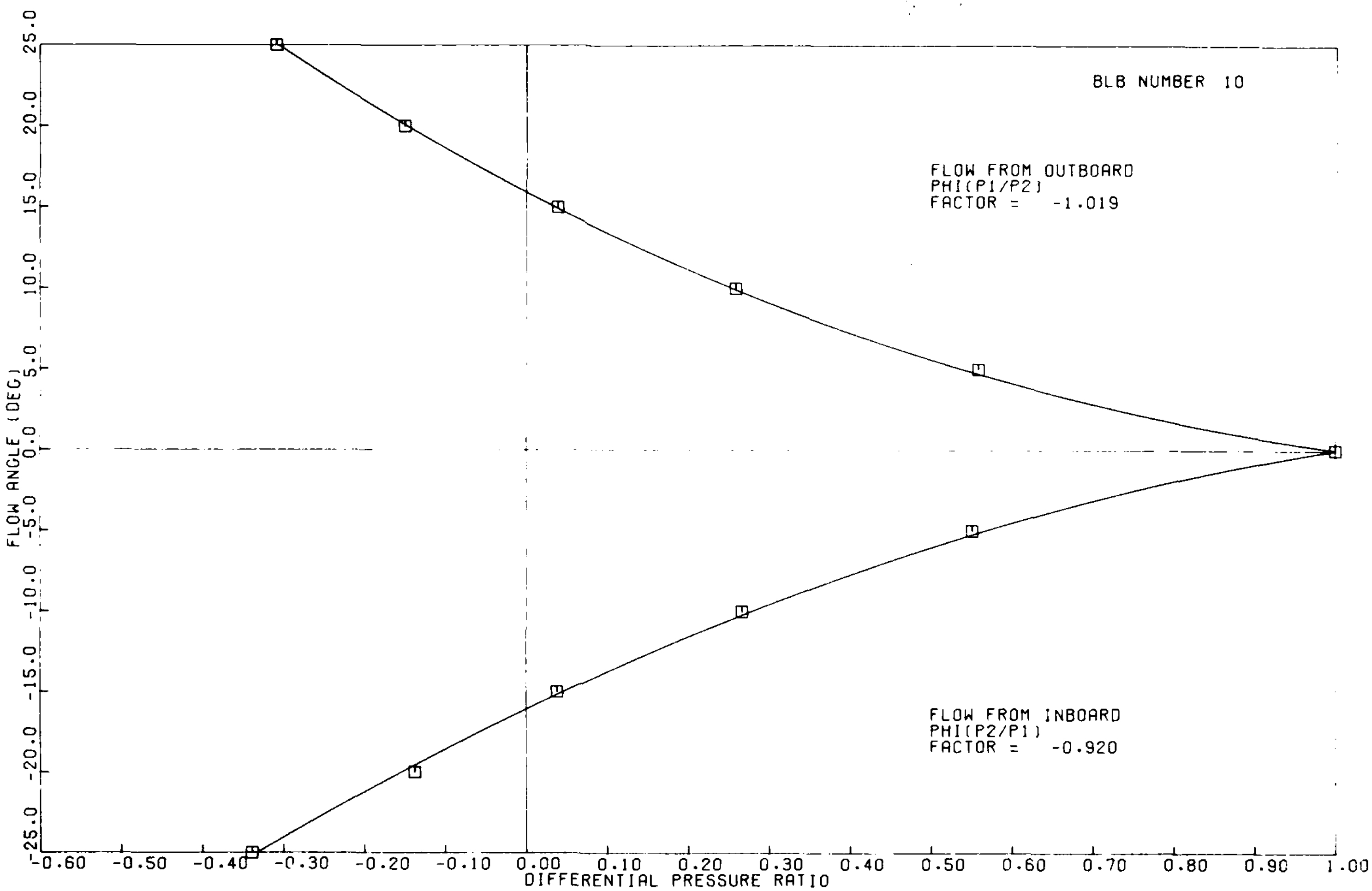


Figure 68. Calibration of BLB number 10.

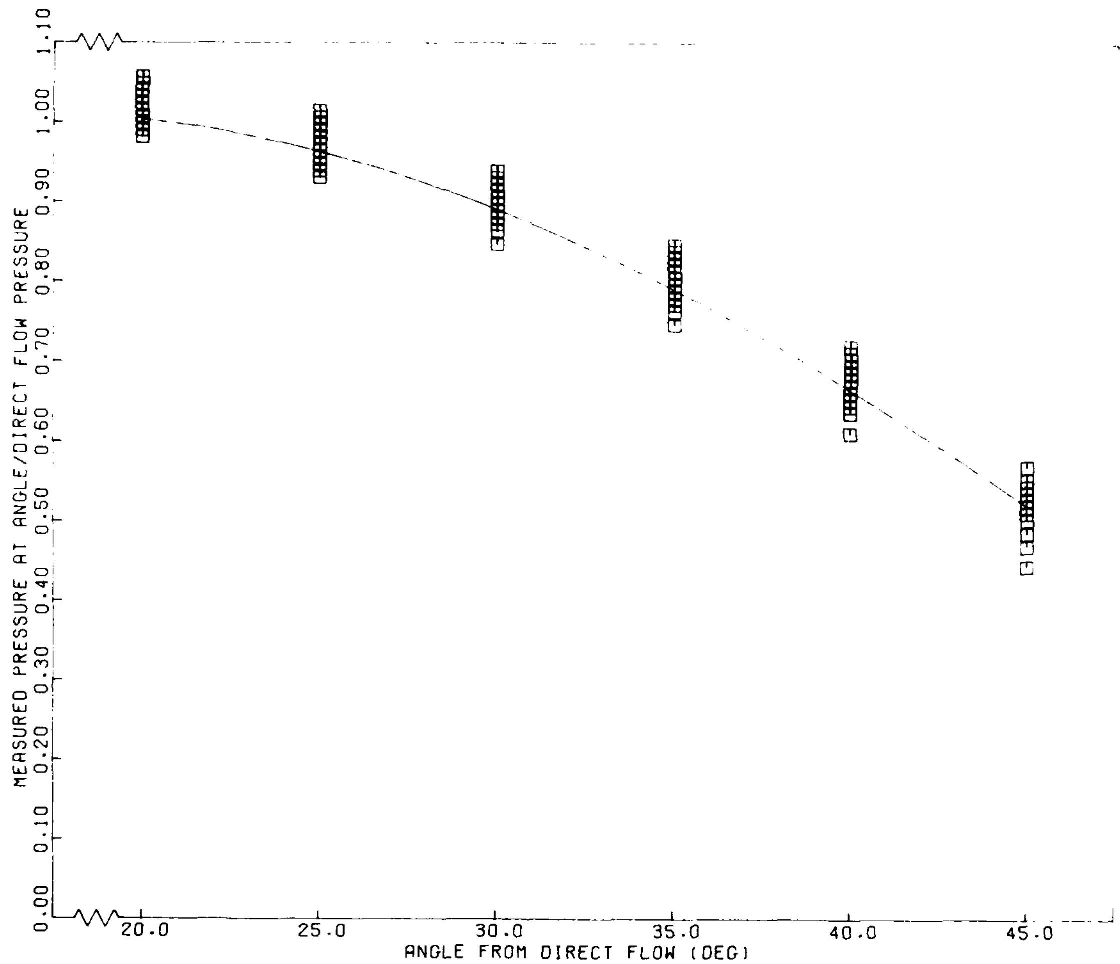


Figure 69. Ratio of direct flow pressure and differential pressure measured at an angle for all BLB sensor tubes.

$$P(t) = 1/2\ \rho(t)\ V^2(t)$$

and

$$P(t) = q(t)/F_P(U(t))$$

then the flow velocity magnitude is

$$V(t) = \sqrt{\frac{2q(t)}{\rho(t)F_P(U(t))}}$$

When $A_B(t)$ exceeds the ±25-degree limitation, the flow is treated as directly into the sensor tube and

$$V(t) = \sqrt{\frac{2q(t)}{\rho(t)}}$$

The BLB flow angle, A_B, is converted to the flow angle relative to the chordline, A_C, by

$$A_C(t) = A_B(t) - (B - 45)$$

The flow magnitude and direction derivations are based on calibrations of the BLBs used in the OLS measurement program. The third-degree polynomial functions F_1, F_2, and F_P are included in the derivation subroutine. The correction factors C_1 and C_2 are stored on the Info File. Processing of BLB data from a different measurement program would require revision of the C_1 and C_2 values stored on the Info File and might require new third-degree polynomials to be installed in the derivation subroutine.

6.2.8 Local Blade Displacement

Local blade displacement is computed using measured accelerometer data. Both displacement and slope are computed from harmonic analysis based on main rotor cycles, and a single harmonic at a time may be printed, plotted, or displayed.

Letting $\ddot{X}(t)$ be measured acceleration in g's, the oscillatory part of the blade motion is

$$\ddot{X}(t) = \sum_{K=1}^{M} (a_K \cos 2\pi K\omega t + b_K \sin 2\pi K\omega t),$$

$$= \sum_{K=1}^{M} C_K \cos(2\pi K\omega t - \phi_k),$$

where $C_K = \sqrt{a_K^2 + b_K^2}$, $\phi_K = \tan^{-1} (\frac{b_K}{a_K})$, and ω is the rotor frequency in hertz. Using the conversion constant C = 386.1 in./(sec^2-g), the displacement for a specific harmonic can be computed as

$$X_K(t) = \frac{CC_K}{(2\pi K\omega)^2} \cos (2\pi K\omega t - \phi_K)$$

in inches. In fact, the harmonic terms C_K and ϕ_K are evaluated using one or more rotor cycles in the same way that the Harmonic Analysis algorithm (Paragraph 6.1.1) can use one or more rotor cycles at the direction of the user. For output, this process creates 256 X(t) values equally spaced in one rotor cycle.

Since the award of this contract, serious deficiencies have been advanced for this method of deriving displacement when the blade is actually rotating. If a rotor speed of 320 RPM and a blade radial station of 260 inches is assumed, the centripetal acceleration at that station is about 760g. Stations nearer the blade root experience proportionately smaller accelerations. If a beamwise accelerometer at a 260-inch station were rotated one degree about a chordwise axis, a measurement of approximately 13g from the 760g centripetal acceleration would be registered. An acceleration amplitude of 13g corresponds to a blade displacement amplitude of about 4.5 inches at 1/rev. Hence, the blade displacement derivation should be used for sensors mounted on a rotating blade with caution.

6.2.9 Local Blade Slope

Local blade slope is interpreted as the derivative of local blade displacements taken along the blade radius for a fixed azimuth. The blade slope algorithm takes as input derived blade displacements for several radial stations along the blade. The slope at each radial station is derived from the quadratic polynomial defined by the displacement at that station and the displacements from the adjacent stations. This process is repeated for every azimuth position represented in the input displacement data. One slope value is generated for every displacement value input.

6.2.10 Density Altitude

Density altitude is computed using the relation

$$H_D = \frac{1-\sigma^{.2349569}}{6.87535(10^{-6})}$$

where σ is air density ratio as described in Paragraph 6.2.2 and the empirical constants are as suggested in Reference 16.

6.2.11 Blade Static Pressure Coefficient

The blade static pressure coefficient, C_p, is computed using the measured parameters blade absolute pressure (psia), boom system static pressure (psia), outside air temperature (°C), and rotor azimuth. In addition, the derived parameters, main rotor RPM, and vehicle true airspeed are required. C_p is computed from

$$C_p = \frac{P_m(t) - P_s(t)}{1/2\rho v^2(t)} K_0 \text{ , where}$$

$P_m(t)$ is blade absolute pressure, $P_s(t)$ is boom system static pressure, ρ is the air density as described in Paragraph 6.2.2, K_0 is the conversion factor 144 in.^{2}/ft^2, and v is the blade station speed given by

$$v(t) = R_{PM}(t)K_1 r + K_2 v_T(t) \sin\psi, \text{ where}$$

$R_{PM}(t)$ is rotor RPM, r is the blade station radial position, K_1 is conversion from revolution/min to radians/sec (i.e.,

---

[16] Eugene P. Bartlett, First Lieutenant USAF, PERFORMANCE FLIGHT TEST HANDBOOK - PART II, AFFTC-TN-59-22 (ASTIA Document Number AD215865), July 1959.

$2\pi/60$), K_2 is conversion from knots to ft/sec (1.688 ft/sec/knot), $v_T(t)$ is vehicle true airspeed and ψ is the rotor azimuth angle.

6.2.12 Blade Normal Force Coefficient

The blade normal force coefficient, C_N, is computed for a fixed radial station and azimuth position from C_p values (Paragraph 6.2.11) for that station. In particular

$$C_N = - \int_0^1 C_p(x,y_u)dx + \int_0^1 C_p(x,y_L)dx$$

where x is the normalized chord station, y_u is the normalized blade surface coordinate in the direction perpendicular to the chordline on the upper blade surface, and y_L is the normalized blade surface coordinate in the direction perpendicular to the chordline on the lower blade surface. For a given chord profile, y_u and y_L are functions of x.

For computational purposes, a finite number of C_p values are available on the upper and lower surfaces of the blade. The algorithm performs the integrations using the trapezoidal method. A C_p value for the x = 0 position is generated by linear extrapolation from the two closest sensors on both the upper and lower surfaces. The two resultant values are then averaged to arrive at a single C_p value for x = 0. A C_p value is also generated for the trailing edge in the same manner using the two sensors closest to the trailing edge for each surface.

6.2.13 Blade Chordwise Force Coefficient

The blade chordwise force coefficient, C_C, is computed for a fixed radial station and azimuth position from C_p values (Paragraph 6.2.11) for that station. In particular,

$$C_C = \int_0^1 C_p(x,y_u)\, y_u'\, dx - \int_0^1 C_p(x,y_L) y_L'\, dx$$

where x, y_u and y_L are described in Paragraph 6.2.12 and y_u' and y_L' are derivatives with respect to x.

For computational purposes, the integrations are performed using the trapezoidal method. Values of C_p for $x = 0$ and $x = 1$ are estimated using the method described in Paragraph 6.2.12.

6.2.14 Blade Pitching Moment Coefficient

The blade pitching moment coefficient, C_M, is computed with respect to the quarter chord position for a fixed radial station and azimuth position from C_p values (Paragraph 6.2.11) for that station. In particular,

$$C_M = - \int_o^1 (1/4-x)\, C_p(x,y_u)dx + \int_o^1 y_u C_p(x,y_u) y_u' dx$$

$$+ \int_o^1 (1/4-x) C_p(x,y_L)dx - \int_o^1 y_L C_p(x,y_L) Y_L' dx$$

where x, y_u and y_L are described in Paragraph 6.2.12 and y_u' and y_L' are derivatives with respect to x.

For computational purposes, the integrations are performed using the trapezoidal method. Values of C_p for $x = 0$ and $x = 1$ are estimated using the method described in Paragraph 6.2.12.

7. REFERENCES

1. Shockey, Gerald A.; Williamson, Joe W.; and Cox, Charles R., AH-1G HELICOPTER AERODYNAMIC AND STRUCTURAL LOADS SURVEY, Bell Helicopter Textron, USAAMRDL Technical Report 76-39, Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Va., February 1977, AD A036910.

2. Philbrick, Richard B. and Eubanks, Alfred L., OPERATIONAL LOADS SURVEY - DATA MANAGEMENT SYSTEM, Bell Helicopter Textron, USARTL Technical Report 78-52A and 52B, Applied Technology Laboratory, Fort Eustis, Virginia, January 1979, ADA065129 and ADA065270.

3. Van Gaasbeek, James R., and Hsieh, P. Y., ROTORCRAFT FLIGHT SIMULATION PROGRAM C81 WITH DATAMAP INTERFACE, Volumes I and II, Bell Helicopter Textron, USAAVRADCOM Technical Report 80-D-38A and 80-D-38B, Applied Technology Laboratory, U. S. Army Research and Technology Laboratories, Fort Eustis, Virginia,

4. Ralston, A., and Wilf, H., MATHEMATICAL METHODS FOR DIGITAL COMPUTERS, John Wiley and Sons, New York, 1960, Chapter 24.

5. Eubanks, A. L., FILTER DESIGN AND ANALYSIS WITH APPLICATIONS TO DISCRETE DATA, Bell Helicopter Textron Report 299-099-889, Fort Worth, Texas, 15 August 1977.

6. Yen, J. G.; Viswanathan, S.; and Matthys, C. G., FLIGHT FLUTTER TESTING OF ROTARY WING AIRCRAFT USING A CONTROL SYSTEM OSCILLATION TECHNIQUE, NASA Symposium on Flutter Testing Techniques, Flight Research Center, Edwards AFB, Calif., October 9-10, 1975.

7. Papoulis, A., PROBABILITY, RANDOM VARIABLES, AND STOCHASTIC PROCESSES, McGraw-Hill, Inc., New York, 1965, Chapters 9 and 10.

8. Stearns, S. D., DIGITAL SIGNAL ANALYSIS, Hayden Book Company, Inc., Rochelle Park, New Jersey, 1975, Chapter 14.

9. Deutsch, R., SYSTEM ANALYSIS TECHNIQUES, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969, Chapter 5.

10. Bendat, J. S., and Piersol, A. G., MEASUREMENT AND ANALYSIS OF RANDOM DATA, John Wiley & Sons, Inc., New York, 1966, Chapter 5.

11. Roth, P. R., EFFECTIVE MEASUREMENTS USING DIGITAL SIGNAL ANALYSIS, IEEE Spectrum, April 1971, pp. 62-70.

12. Carnahan, B.; Luther, H. A.; and Wilkes, J. O., APPLIED NUMERICAL METHODS, John Wiley & Sons, Inc., New York, 1969, Chapter 8.

13. Federal Aviation Regulations, Part 36, Appendix B. Department of Transportation, Federal Aviation Administration, April 1978.

14. RECOMMENDATIONS FOR SOUND LEVEL METERS, International Electrotechnical Commission, Geneva, Switzerland, 1961.

15. Instructions and Applications, IMPULSE PRECISION SOUND LEVEL METER TYPE 2204, Brüel & Kjaer, Copenhagen, Denmark, July 1969.

16. Bartlett, Eugene P., First Lieutenant USAF, PERFORMANCE FLIGHT TEST HANDBOOK - PART II, AFFTC-TN-59-22 (ASTIA Document Number AD215865), July 1959.

APPENDIX A

PROCESSING ERROR NUMBERS

Many user errors can be detected as a Processing Program command step is being entered by the user. Other errors, as well as possible program errors, cannot be detected until processing proceeds. When the program detects an error in processing, an error number is listed and processing of the command step is terminated. For certain errors, a diagnostic message is generated along with the error number. All of these error numbers are cataloged in this appendix. Listed along with each number is the subroutine that detected the error, an indicator '(P)', '(U)' or '(E)', and a short explanation of the error. The indicator (P) specifies a probable program error, (U) indicates a probable user error, and (E) indicates that the source of the error is uncertain.

Currently there are no known processing sequences that would generate an error number with a (P) indicator. However, in a program of this size and complexity, it is impossible to test every conceivable path of execution with every possible data input. In addition, modification of the program to include additional analyses, derivations, output capabilities, or user interface commands could introduce processing errors that would be more easily isolated by those error diagnostics.

NUMBER	ROUTINE	INDICATOR	PROBLEM
1	RMS	(P)	Routine called with an incorrect file number.
2	RMS	(U)	Record wanted outside data area.
3	RMS	(E)	Record wanted outside file area.
4	RMS	(P)	Record size too large.
5	WMS	(P)	Routine called with incorrect file number.
6	WMS	(U)	Record position outside allowed area.
7	WMS	(E)	Record position outside file area.
8	WMS	(P)	Record size too large.
9	FMS	(P)	Routine called with incorrect file number.
10	FMS	(U)	Record wanted outside data area.
11	FMS	(E)	Record wanted outside file area.
101	PROSET	(P)	Bad user interface value specified.
102	PROSET	(P)	Incorrect 'ANALYZE' interface specification.
103	PROSET	(P)	Incorrect 'DERIVE' interface specification.
121 122 123 124	COMPSC COMPSC COMPSC COMPSC	(E) (E) (E) (E)	May be attempting to read a scratch file which has never had data written into it. If data have been written to the scratch file previously, then these numbers could indicate a program error.
130	COMPSC	(P)	Incorrect user interface value specified.
135	COMPSC	(U)	A row or column was specified for extraction from a scratch file which does not exist on that file.

151 152 153 154	COMPSC COMPSC COMPSC COMPSC	(P) (P) (P) (P)	Error reading attached parameter data from a scratch file.
160	COMPSC	(U)	Attached parameter data (azimuth, air-speed, RPM, OAT or static pressure) required for a process are not available on the scratch file being read.
165	COMPSC	(U)	A first-dimension scale is specified which conflicts with the available scales.
170	COMPSC	(U)	A first dimension parameter value has been specified which does not occur in the input scratch file.
175	COMPSC	(U)	Type of parameter required for a derivation is not present on the specified scratch file.
180	COMPSC	(U)	A specified first dimension parameter value is beyond the range of stored values.
190	GETDAT	(U)	Specified input data would overflow the available storage area.
195	GETDAT	(U)	The initial data stream required for the process specified is not available.
200	FINDT	(P)	Routine called with illegal parameter specified.

NUMBER	ROUTINE	INDICATOR	PROBLEM
201	FINDT	(U)	A parameter value used to specify input data does not occur for the available data.
202	FINDT	(U)	A parameter used to specify input data is not available.
210	GET1	(U)	Specified input data would overflow the available storage area.
220	GET1	(U)	Insufficient azimuth data to specify number of cycles required for input.
230	INPSET	(P)	Incorrect user interface value specified.
240	INPSET	(U)	Specified counter is not present on the Master File partition.
270	INFSCR	(U)	No data present in scratch file specified for input.
275	INFSCR	(U)	Sample intervals for cross process inputs do not match.
280	RTRVSC	(E)	Data to be read from scratch file would overflow program storage.
304	AMPSET	(U)	An insufficient number of points were provided to generate an amplitude spectrum.
325	TSAV1	(P)	Improper file specified for random access write

NUMBER	ROUTINE	INDICATOR	PROBLEM
326 327	TSAV1 TSAV1	(U) (U)	Processed data over- flows temporary scratch file area.
328	TSAV1	(P)	Improper record size specified for random access write to tem- porary scratch file.
345	TSAV1	(P)	Improper file specified for random access write to temporary scratch file.
346 347	TSAV1 TSAV1	(U) (U)	Processed data over flows SCF1 or SCF2.
348	TSAV1	(P)	Improper record size specified for random access write to SCF1 or SCF2.
350	PRO1	(P)	Bad user interface value specified.
365	TSAV2	(P)	Improper file specified for random access write.
366 367	TSAV2 TSAV2	(U) (U)	Processed data over- flows SCF1 or SCF2.
368	TSAV2	(P)	Improper record size specified for random access write to SCF1 or SCF2.
375	TSAV2	(P)	Improper file specified for random access write to temporary scratch.
376 377	TSAV2 TSAV2	(U) (U)	Processed data over flows SCF1 or SCF2.
378	TSAV2	(P)	Improper record size specified for random access write to tempo- rary scratch file.

NUMBER	ROUTINE	INDICATOR	PROBLEM
380	TSAV2	(P)	Incorrect process output sequence.
390	TSAV2	(U)	Column scale specified for output with single column element available.
401	MULTPL	(P)	Incorrect file specified in reading data from direct access temporary scratch.
402 403	MULTPL MULTPL	(P) (P)	Incorrect record number specified for read from direct access temporary scratch file.
404	MULTPL	(P)	Incorrect record size specified for read from direct access temporary scratch file.
430	SINGPL	(U)	Attached parameter values are missing for generation of output scale.
440	DISPOS	(U)	APLOT specified when a plot scale generated by MPLOT is not already present.
460	OUTSET	(P)	Incorrect user interface value specified.
470	OUTSET	(U)	Plot output is specified when the process output will consist of a single point.
475	OUTSET	(U)	The output mode selected provides for a different number of independent variable dimensions than the output data will possess.

NUMBER	ROUTINE	INDICATOR	PROBLEM
480	ALLATT	(E)	Error reading measured data to generate OAT, true airspeed or static pressure.
490	AZIMTH	(U)	Insufficient azimuth data are available to define a stream specified in cycles.
495	AZIMTH	(U)	Attached parameter storage area overflows.
500	IMPATT	(P)	Incorrect value in Info record for data on the Master File.
510	ATTGET	(E)	Error generating azimuth times.
520	ATTGEN	(U)	Insufficient airspeed, OAT, or static pressure data for requested display.
530	ATTGEN	(U)	Data generated overflows available program storage.
540	AZMGEN	(U)	Insufficient azimuth data for requested display.
542	AZMGEN	(U)	Generated azimuth values overflow the data storage area.
550	AZMGEN	(U)	Data generated overflows available program storage.
560	POWGEN	(U)	Insufficient azimuth, airspeed, OAT, or static pressure data for the requested derivation.

570	POWGEN	(P)	Error interpolating data.
580	SCALGN	(U)	Insufficient azimuth or airspeed for requested scale.
610	COMPGP	(U)	Specified Info File group name not specified.
620	READGP	(U)	Info File group is incomplete.
630	READGP	(U)	Info File group has line error.
635	COMPGP	(U)	Line error in Info File.
640	READGP	(U)	Info File group has syntax error.
650	COMPGP	(P)	Incorrect specification from user interface.
660	COMPGP	(U)	Row element specified not present in Info File group.
670	COMPGP	(U)	Column element specified not present in Info File group.
680	DFILTR	(U)	Too few data points provided for filtering process.
690	HARMNY	(U)	The input data stream contained two doublerow elements. The Harmonic Analysis process can handle only one doublerow element.
700	HARMNY	(U)	The input data stream does not span a full rotor cycle.

NUMBER	ROUTINE	INDICATOR	PROBLEM
710	HARMNY	(U)	The input data sample rate is too low to calculate the specified harmonics.
740	DSPSET	(U)	A BOTTOM doublerow element was specified for a blade displacement derivation.
752	DSPSET	(U)	Insufficient accelerometer data to generate blade displacement for one complete rotor cycle.
753	DSPSET	(U)	Accelerometer data sample rate is too low to calculate blade displacement for the specified harmonic.
771	DAMPST	(U)	Input data stream for moving block damping is too short or has too low a sample rate.
780	PRCFST	(U)	Total data requirements overflow available program storage.
785	PRCFST	(U)	Insufficient attached parameter data for Cp derivation.
790	FLWSET	(U)	Input data stream overflows available program storage.
795	FLWSET	(U)	Insufficient attached parameter data for Blade Local Flow Magnitude or Direction derivation.
800	YSFRST	(E)	Generated contour or surface plot matrix will overflow available storage.

NUMBER	ROUTINE	INDICATOR	PROBLEM
805	YSFRST	(U)	Intermediate data matrix will overflow available storage in generating contour or surface plot.
811	YSFRST	(P)	Temporary scratch file read error while generating a data matrix for a surface or contour plot.
812	YSFRST	(P)	
813	YSFRST	(P)	
814	YSFRST	(P)	
825	TSAV3	(E)	SCF1 or SCF2 write error. Possible over flow.
826	TSAV3	(E)	
827	TSAV3	(E)	
828	TSAV3	(E)	
831	TSAV3	(P)	Temporary scratch file read error.
832	TSAV3	(P)	
833	TSAV3	(P)	
834	TSAV3	(P)	
835	TSAV3	(P)	Temporary scratch file write error.
836	TSAV3	(P)	
837	TSAV3	(P)	
838	TSAV3	(P)	
850	CONSET	(U)	Cylindrical format 3-D plot requested with first independent variable other than azimuth.
860	CONREC	(U)	3-D plot requested in nongraphics mode.
871	GETEMP	(P)	Temporary scratch file read error.
872	GETEMP	(P)	
873	GETEMP	(P)	
874	GETEMP	(P)	
880	GETEMP	(U)	Input data overflows available program storage.

NUMBER	ROUTINE	INDICATOR	PROBLEM
921 922 923 924	CYAVST CYAVST CYAVST CYAVST	(P) (P) (P) (P)	Program errors in generating cycle averaged data.
940	CYAVST	(U)	Complete cycle of data not available.
950	MINXST	(P)	Cycle start time does not match start of data.
960	MINXST	(U)	Complete cycle of data not available.
970	SLOPST	(U)	Data overflows available program storage.
1005	CONVCK	(P)	Input error reading unit conversion lines of Info File.
1010	CONVCK	(U)	Line error in a unit conversion line of Info File.
1020	ACORST	(U)	Too few points for auto-correlation.
1025	XCORST	(U)	Too few points for cross-correlation.
1030	ACORST	(U)	Too many points to perform auto-correlation with available storage.
1035	XCORST	(U)	Too many points to perform cross-correlation with available storage.
1040	ACONST	(P)	Cycle-averaged number of points is not a power of two.
1045	XCORST	(P)	Cycle-averaged number of points is not a power of two.

NUMBER	ROUTINE	INDICATOR	PROBLEM
1050	ADENST	(U)	Too few points input for auto spectral density.
1080	COHRST	(U)	Too few points input for Coherence Function calculation.
1090	XDEMST	(U)	Too few points input for cross-spectral density.
110	RESPST	(U)	Too few points input for frequency response function.
1110	COHRST	(U)	Single input record for coherence function calculation. Ensemble averaging is required.
1120	NARRST	(U)	Sample interval too wide for acoustic analysis.
1140	NARRST	(U)	Illegal band width for narrow band analysis. (too wide or too narrow)
1160	OCTVST	(U)	Sample interval too wide for acoustic analysis.
1170	OCTVST	(U)	No octave or third octave levels could be generated.
1180	DBWTST	(U)	Sample interval too wide for acoustic analysis.
1190	DBWTST	(U)	Record too short for level integration.
2000	PNLDST	(U)	Sample interval too long for PNL calculation.
2010	PNLCAL	(U)	Record too short for PNL calculation.

LMED

8